Magnifying What Matters: Attention-Guided Adaptive Rendering for Visual Text Comprehension

Shenglai Zeng[1] **Qirui Wang**[2] **Kai Guo**[1†] **Xinnan Dai**[1] **Xianxuan Long**[1] **Hui Liu**[1]
[1]Michigan State University [2]Xi'an Jiaotong University

Abstract

Visual Text Comprehension (VTC) renders text into images for a vision–language model (VLM) to read, sidestepping LLM context-window limits and powering applications from long-page OCR to multi-page memory QA. Yet existing VTC pipelines treat rendering and layout as a fixed, content-agnostic preprocessing step, and offer little mechanistic understanding of how VLMs internally process visualized text. Through a focused empirical study on VTC QA tasks, we reveal that VLMs exhibit a *localization-without-utilization* regime: evidence-localizing attention emerges sharply in the middle-to-late layers and is largely decoupled from answer correctness, yet simply enlarging the localized spans on the rendered page recovers a large fraction of the failures. Building on these observations, we propose **AGAR** (Attention-Guided Adaptive Rendering), a training-free, model-agnostic method that leverages a VLM's own middle-to-late layer attention to identify the top-K important visual patches, maps them back to word spans, and re-renders the page with those spans enlarged before re-inferring the answer. Extensive experiments across nine VTC benchmarks (short-form, long-context, and multi-page memory QA) and four VLM backbones show that AGAR (i) consistently improves off-the-shelf VLMs as a plug-and-play enhancement, (ii) composes with VLM post-training to yield further gains, and (iii) remains robust under both visual- and text-side input degradation.[1]

1 Introduction

The context window of large language models (LLMs) remains a fundamental bottleneck that limits how much text a model can process in a single pass. A recently emerging paradigm, *Visual Text Comprehension* (VTC) or *optical context compression*, sidesteps this bottleneck by *rendering* text into images and letting a vision–language model (VLM) read it directly [29, 30, 5]. A single visual patch can pack many characters, yielding $3-20\times$ token-level compression and unlocking applications such as long-page OCR [29], long-context visual question answering [5], vision-based long-term memory for agents [23, 7], reasoning-chain compression [28], persona-conditioned recommendation [20], and long-context code understanding [37].

Despite this momentum, current VTC pipelines leave two important gaps that remain underexplored. First, most prior work treats rendering and layout as a fixed, content-agnostic step. But typography itself is a useful signal. Textbooks enlarge headings and bold keywords to direct a human reader's eye [21, 10], and similar visual emphasis could possibly route a VLM's limited visual capacity toward task-relevant content. Second, the community has limited *mechanistic* understanding of how VLMs internally process VTC inputs [19, 4]: which layers actually localize task-relevant text, whether failures stem from attending to the wrong region or from mis-reading the right one, and how the model consumes the localized signal. Understanding these internal mechanisms is crucial for diagnosing bottlenecks and designing effective enhancement methods. By contrast, in adjacent

†Corresponding author: guokai1@msu.edu.
[1]Code available at https://github.com/AGAR-NeurIPS2026/AGAR-NeurlPS2026.

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

areas such as evidence-based question answering and traditional VQA, prior work has shown that the internal representations and attention patterns of LLMs already contain rich, exploitable signals about what the model is truly attending to [32, 34, 16, 17], suggesting that a similar internal lens may be the missing tool for VTC as well.

To investigate the two gaps above, we conduct a comprehensive empirical study on VTC QA tasks (§3) that examines how a VLM reads long rendered text and where its failures come from, looking at both its attention patterns and its answer quality. We make three key observations. **First**, evidence-localizing attention emerges sharply in the middle-to-late layers and is carried by a substantial fraction of heads in those layers, consistently across backbones. **Second**, this localization is largely decoupled from correctness: incorrect samples attend to the right evidence nearly as well as correct ones, placing VLMs in a *localization-without-utilization* regime. **Third**, on failing samples, enlarging the ground-truth evidence on the rendered page recovers a large fraction of errors with a high fix-to-break ratio. These findings suggest that, at least to a large extent, the bottleneck lies in *utilizing* the localized evidence rather than discovering it.

Inspired by the above observations, we develop **AGAR** (Attention-Guided Adaptive Rendering), a training-free, model-agnostic wrapper around any VLM that exposes layer-wise attentions. AGAR first leverages middle-to-late layer attention scores to identify the top-K important visual patches and maps them back to word spans; it then re-renders the page with those spans enlarged and re-infers the answer. Extensive experiments across nine VTC benchmarks spanning short-form, long-context, and multi-page memory QA, and four VLM backbones (Qwen3-VL-8B [24, 2], InternVL3.5-8B [27], GLM-4.1V-9B-Thinking [9], and Glyph [5]), show that AGAR is **(i)** training-free, raising VTC accuracy on off-the-shelf VLMs as a plug-and-play enhancement; **(ii)** complementary to post-training, yielding additional gains when its attention-guided magnification is incorporated into supervised fine-tuning; and **(iii)** robust to both visual-side (lower resolution, font/style perturbations) and text-side (distractor injection, noisy contexts) input degradation.

In summary, our contributions are:

- We identify a *localization-without-utilization* regime in VTC: VLMs already attend to the right evidence in the middle-to-late layers even when the answer is incorrect, yet enlarging those spans on the rendered page recovers many failures, pinpointing evidence *utilization* as a key bottleneck.
- We propose **AGAR**, an attention-guided adaptive rendering method that closes this gap by using the model's own attention to select evidence spans and enlarging them in a re-rendered page, with no weight or prompt changes.
- Extensive experiments across 9 benchmarks and 12 subtasks on four VLM backbones show that AGAR (a) improves off-the-shelf VLMs in a training-free manner, (b) composes with VLM post-training to yield further gains, and (c) is robust to both visual- and text-side input degradation.

2 Related Works

2.1 Visual Text Comprehension

Visual Text Comprehension (VTC), or optical context compression, sidesteps the LLM context-window bottleneck by rendering text into images and letting a VLM read it directly [29, 30, 5, 36, 14]. Building on this paradigm, a growing body of work targets downstream applications including vision-based long-term agent memory [23, 7], long reasoning-chain compression [28], long-context code understanding [37], and generative recommendation [20]. Most of these works, however, either treat the rendering pipeline as a fixed, content-agnostic preprocessing step, or train a dedicated model to pre-compress the context end-to-end [23]; a clear gap remains in understanding how VLMs internally process VTC inputs and in exploiting such mechanistic signals to drive context-aware rendering.

2.2 Utilization of (V)LLMs' Internal Signals

An emerging research direction focuses on utilizing (V)LLMs' internal signals to understand how they process input contexts. On the LLM side, Zeng et al. [33] use hidden representations to identify high-level concepts in RAG systems such as context helpfulness; Liu et al. [16] use last-token attention to surface important evidence spans in text QA; and Zeng et al. [34] exploit attention scores to locate critical personalization signals in long user histories. On the VLM side, prior work in classical VQA has shown that attention-related signals can be used to amplify task-relevant visual regions

and suppress irrelevant ones, via Grad-CAM-style attribution [22, 1], attention-guided cropping [35], attention-driven masking [15], and brightness/contrast modulation [17]. VTC, however, lies at the intersection of these two regimes—visual inputs that encode dense, structured text rather than natural-image scenes or pure-text tokens—and has not been systematically studied through this internal-signal lens.

3 Preliminary Studies

Before describing our method, we conduct a focused empirical study across four open-weight VLMs (Qwen3-VL-8B, InternVL3.5-8B, GLM-4.1V-9B-Thinking, and Glyph[2]) to understand *how* a VLM reads long rendered text and where its failures come from. After introducing notation and metrics (§3.1), we organize the study around three research questions that progressively dissect the gap between *seeing* and *using* evidence: **RQ1** (§3.2) asks *where* evidence-localizing attention emerges inside a VLM; **RQ2** (§3.3) asks whether VTC failures are themselves localization failures; **RQ3** (§3.4) asks how to help the model use the evidence it already localizes. Each subsection ends with a boxed observation; together they motivate the design of our method (§4).

3.1 Problem Formulation

Visual Text Comprehension (VTC). Given a long textual context T rendered into one or more images $\mathcal{I} = \{I_1, \ldots, I_{|\mathcal{I}|}\}$ and a natural-language question q, a vision–language model (VLM) $\mathcal{M}$ produces an answer $\hat{a} = \mathcal{M}(\mathcal{I}, q)$. We assume that each ground-truth answer $a^\star$ is supported by a set of *evidence spans* $\mathcal{E} \subseteq T$, i.e., the key evidence in T supporting $a^\star$ (provided as explicit annotations in the QA datasets we use).

Image patches and evidence labels. The VLM's visual encoder partitions $\mathcal{I}$ into a grid of patches and emits N visual tokens $\mathcal{P} = \{p_1, \ldots, p_N\}$, one per patch.[3] Using the renderer's word–bounding-box map together with $\mathcal{E}$, each patch is tagged as *evidence* or *non-evidence*: $y_i = 1$ if patch p_i overlaps any rendered region of $\mathcal{E}$, otherwise $y_i = 0$. The evidence set is denoted as $\mathcal{P}_\mathcal{E} = \{p_i : y_i = 1\}$.

Patch-level attention scores. Attention is extracted from the last query token q_{last} (the position that drives next-token prediction) to image patches. Let L denote the number of decoder layers and H the heads per layer. For each layer ℓ and head h, the per-head attention vector is

$$\boldsymbol{\alpha}^{(\ell,h)} \;=\; \big[\, A^{(\ell,h)}_{q_{\text{last}},\,p_1},\; \ldots,\; A^{(\ell,h)}_{q_{\text{last}},\,p_N} \,\big] \;\in\; \mathbb{R}^N,$$

where $A^{(\ell,h)}$ is the (post-softmax) self-attention map at layer ℓ, head h. Two aggregated attention vectors are used downstream. (i) The *head-mean attention vector* per layer, $\bar{\boldsymbol{\alpha}}^{(\ell)} = \frac{1}{H}\sum_{h=1}^{H} \boldsymbol{\alpha}^{(\ell,h)}$. (ii) The *layer-range aggregated attention vector* over a contiguous interval $\mathcal{L}_{[a,b]} = \{\ell : \lceil aL \rceil \le \ell \le \lfloor bL \rfloor\}$ with $0 \le a < b \le 1$,

$$\tilde{\boldsymbol{\alpha}}_{[a,b]} \;=\; \frac{1}{|\mathcal{L}_{[a,b]}|} \sum_{\ell \in \mathcal{L}_{[a,b]}} \bar{\boldsymbol{\alpha}}^{(\ell)}.$$

Localization quality (NDCG). Given an attention score vector $\mathbf{s} \in \mathbb{R}^N$ and the binary evidence labels $\mathbf{y}$, we measure its evidence-localization quality using Normalized Discounted Cumulative Gain [11]. Sorting $\mathcal{P}$ by $\mathbf{s}$ (NDCG) in descending order with $\pi(r)$ denoting the patch at rank r, we define $\mathrm{DCG}(\mathbf{s}, \mathbf{y}) = \sum_{r=1}^{N} y_{\pi(r)} / \log_2(r+1)$ and $\mathrm{NDCG}(\mathbf{s}, \mathbf{y}) = \mathrm{DCG}(\mathbf{s}, \mathbf{y})/\mathrm{IDCG}(\mathbf{y})$, where IDCG is the DCG of the ideal ranking that places all $|\mathcal{P}_\mathcal{E}|$ positives first. NDCG $= 1$ corresponds to a perfect localization that ranks every evidence patch above every non-evidence one. **A higher NDCG therefore means the attention scores $\mathbf{s}$ assign larger weights to the patches that actually support the answer, separating them from non-evidence ones.**

3.2 RQ1: In the VTC setting, where does evidence-localizing attention emerge inside a VLM?

To investigate *where* a VLM concentrates its visual attention when reading rendered text, we measure two complementary signals on HotpotQA [31] (results on other datasets are deferred to App. D and E):

[2] Glyph is post-trained on GLM-4.1V-9B-Thinking for VTC.

[3] A "patch" is the image region of one visual token (possibly multiple ViT sub-patches after spatial merging).

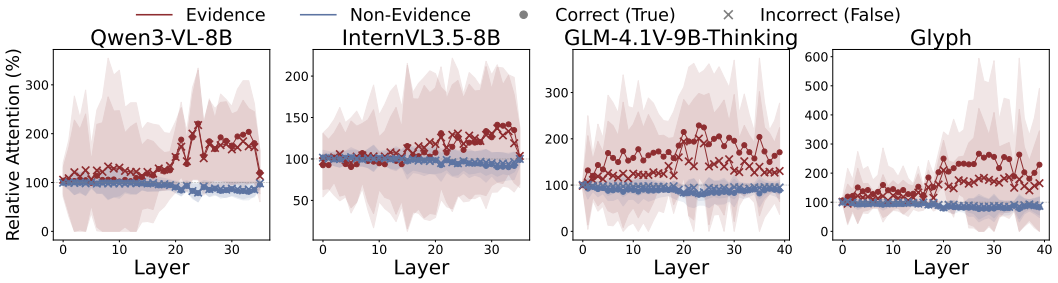


Figure 1: Relative attention (%) per layer on HotpotQA. Red/blue lines: attention on Evidence/Non-Evidence tokens. ●/×: correctly/incorrectly answered samples.

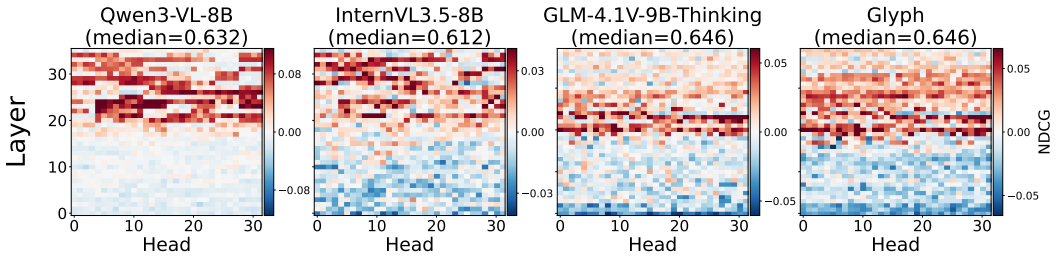


Figure 2: Per-head NDCG of last-token attention to evidence tokens on HotpotQA. Colors encode deviation from the per-model median NDCG.

(i) the average *relative attention* (%) on evidence vs. non-evidence image tokens at each decoder layer (Fig. 1), where 100% denotes the layer-wise mean attention so values above 100% indicate above-average focus; and (ii) the per-head localization quality NDCG defined in §3.1, visualized as a layer×head heatmap (Fig. 2).

Layer-wise emergence of evidence focus. Across all four model backbones, the curves in Fig. 1 share a common pattern. In the early layers (roughly the first $\sim L/2$), the relative attention on evidence and non-evidence tokens is statistically indistinguishable, both hovering around the layer mean of 100%. Starting from the middle layers, evidence attention rises sharply, opens a substantial gap above non-evidence attention, and *remains elevated through the last layer*. The shift is consistent regardless of architecture (Qwen, InternVL, GLM-4.1V) or post-training (GLM vs. Glyph). A plausible explanation is that the early layers are still performing generic visual-feature aggregation (e.g. character/glyph recognition over the rendered page), while the late-middle layers begin task-conditioned reading: once the question token has had sufficient depth to build up a query, the model selectively routes additional attention to the spans that actually answer it. The same trend also appears on other datasets (Appendix D, Fig. 8).

Per-head structure of localization. Fig. 2 shows that heads in the lower half of the network have very limited localization ability, while the upper half contains a population of heads whose NDCG sits well above the median. Evidence localization is thus not a few isolated heads at random depths, but a *population* that activates together at middle-to-later layers. The same pattern appears on other datasets (App. E, Fig. 9).

Together, these observations identify the middle-to-late layers as a contiguous band where evidence-localization information is concentrated, which helps us decide to use the localization signal from the upper half of layers in Section 4.[4]

Observation 1. Evidence-focused attention emerges at middle-to-late layers, populated by a band of evidence-attending heads, consistently across the four VLMs we examine.

[4]See Fig. 10 for per-head NDCG vs. head rank and Table 11 for layer-range aggregated NDCG.

3.3 RQ2: Are VTC failures explained by failed evidence localization?

A natural follow-up question is whether VTC failures are caused by localization failures: when the model answers incorrectly, has it simply failed to look at the right evidence? Fig. 1 splits the per-layer relative attention curves by per-sample correctness (SubEM $= 1$ vs. SubEM $= 0$[5]); circles mark correct samples, crosses mark incorrect. Strikingly, the two subsets share the same overall trajectory. Even on incorrectly answered samples, the relative attention on evidence tokens rises sharply in the middle-to-late layers and stays well above the non-evidence baseline through the last layer. A modest gap between the Correct and Incorrect curves does exist, most visible on the GLM-family models (GLM-4.1V-9B-Thinking and Glyph), where Incorrect samples sit slightly lower. Even so, on Incorrect samples the attention on evidence tokens remains obviously higher than on non-evidence tokens. In other words, the model is already *looking at* the right evidence on a substantial fraction of the samples it ultimately gets wrong. This points to a *localization-without-utilization* regime: the model correctly identifies which patches carry the evidence, yet does not always translate that localized signal into the correct answer. Localization quality is therefore a necessary but not a sufficient condition for VTC accuracy. Interventions that target the *utilization* side, such as making the localized evidence easier for the model to consume, have the potential to mitigate this gap (RQ3).

Observation 2. Localization quality is largely decoupled from answer correctness: incorrect samples attend to the right evidence almost as well as correct ones. VLMs in VTC are in a *localization-without-utilization* regime.

3.4 RQ3: How can we help the model better utilize the evidence it already localizes?

Following the question raised by RQ2 (how to help the model better *utilize* the evidence it already attends to), we draw inspiration from how a human reader handles a long passage. Once the salient sentences are spotted, a natural reflex is to *look closer* at them, i.e., to magnify the relevant text. We ask whether the same intervention helps a VLM. To isolate the effect from any localization error, we use the ground-truth spans $\mathcal{E}$ as an oracle and re-render each page with characters in $\mathcal{E}$ drawn at $1.5\times$ the baseline font size, leaving the rest of the page (layout, surrounding distractors, prompt, and model weights) unchanged. Re-querying Qwen3-VL-8B on $1{,}000$ random HotpotQA samples (Table 1), we observe that 21.9% of originally incorrect samples are recovered ($\mathrm{F}\to\mathrm{T}$) while only 5.5% of correct ones are broken ($\mathrm{T}\to\mathrm{F}$), giving a fix:break ratio of $\approx 4{:}1$ and a ~ 16-point SubEM gain—all from changing only the on-page typography. Magnifying the important information thus has clear potential to help the model better utilize the evidence it already localizes.

Table 1: SubEM transition matrix on HotpotQA (Qwen3-VL-8B, 1000 samples).

	Enlarged: T	**Enlarged: F**
Plain: T	26.0% (TT)	5.5% (TF)
Plain: F	21.9% (FT)	46.6% (FF)

Observation 3. Localized magnification substantially recovers VTC failures, indicating its potential to mitigate the *localization-without-utilization* regime identified in Obs. 2.

4 AGAR: Attention-Guided Adaptive Rendering

The three observations in §3 suggest a simple but effective recipe. Obs. 1 indicates *where* to read the model's own localization signal: in the middle-to-late attention layers. Obs. 2 suggests that this localization is, on its own, often not enough to flip a failed sample. Obs. 3 shows that physically magnifying the localized text can recover a substantial fraction of those failures. The natural method is then to chain the three: *use the model's own attention (especially the middle-to-late part) to decide which words to magnify, re-render the page accordingly, and re-query the model*. We call this method **AGAR** (Attention-Guided Adaptive Rendering); a high-level sketch is given in Figure 3 and Algorithm 1. AGAR proceeds in two stages, all without modifying the model weights or the prompt. We follow the notation of §3.1.

[5] SubEM is a proxy for evidence use: if evidence text appears in the answer, it likely was used. §5 adopts standard EM/F1.

Algorithm 1 AGAR: Attention-Guided Adaptive Rendering for VTC.

Require: context T, question q, VLM $\mathcal{M}$, renderer $\mathcal{R}$, layer range $[a, b]$, top-K size K, font scale s_{font}.

1: $(\mathcal{I}^{(0)}, \mathcal{W}^{(0)}) \leftarrow \mathcal{R}(T)$ ▷ baseline render with word boxes

2: $\{\boldsymbol{\alpha}^{(\ell,h)}\} \leftarrow \mathcal{M}\cdot\textsc{Forward}(\mathcal{I}^{(0)}, q;\ \texttt{output_attentions})$

3: $\mathbf{s} \leftarrow \tilde{\boldsymbol{\alpha}}_{[a,b]}$ ▷ Stage 1: late-layer head-mean aggregation

4: $\mathcal{P}^\star \leftarrow \mathrm{TopK}(\mathbf{s}; K);\ \hat{\mathcal{E}} \leftarrow \textsc{Map}(\mathcal{P}^\star, \mathcal{W}^{(0)})$ ▷ select top-K patches; lift to char spans

5: $\mathcal{I}^{(1)} \leftarrow \mathcal{R}(T, \hat{\mathcal{E}}, s_{\text{font}})$ ▷ Stage 2: re-render with $\hat{\mathcal{E}}$ enlarged

6: $\hat{a} \leftarrow \mathcal{M}(\mathcal{I}^{(1)}, q)$ ▷ answer from the magnified page

7: **return** $\hat{a}$

Stage 1 – Attention-Based Evidence Localization. We first render the textual context T at the baseline font size with the renderer $\mathcal{R}$, obtaining the raw image $\mathcal{I}^{(0)}$ together with a word–bounding-box map $\mathcal{W}^{(0)} = \{(w, \beta_w, [c_w^s, c_w^e))\}$ (Alg. 1, line 1) that records, for every rendered word w, its image-space box β_w and the character span $[c_w^s, c_w^e)$ it occupies in T. A single forward pass of the VLM on $(\mathcal{I}^{(0)}, q)$ with `output_attentions=True` yields the layer-wise attentions (Alg. 1, line 2), which we collapse into a per-patch score $\mathbf{s} = \tilde{\boldsymbol{\alpha}}_{[a,b]} \in \mathbb{R}^N$ with $[a, b] = [0.5, 1.0]$ by default (Alg. 1, line 3; Obs. 1). The top-K patches $\mathcal{P}^\star = \mathrm{TopK}(\mathbf{s}; K)$ live in the model's visual grid; since the unit we can actually magnify is a rendered word, we *map* the selection from patches to character spans: $\hat{\mathcal{E}} = \textsc{Map}(\mathcal{P}^\star, \mathcal{W}^{(0)})$ (Alg. 1, line 4). Concretely, MAP marks a word $w \in \mathcal{W}^{(0)}$ as selected whenever its box β_w overlaps any selected patch $p_i \in \mathcal{P}^\star$ (after rescaling β_w to the visual encoder's input resolution), collects the corresponding character spans $[c_w^s, c_w^e)$, and merges overlapping/adjacent spans. The output $\hat{\mathcal{E}} \subseteq T$ is our *predicted* evidence—the attention-only counterpart of the oracle $\mathcal{E}$ used in Obs. 3.

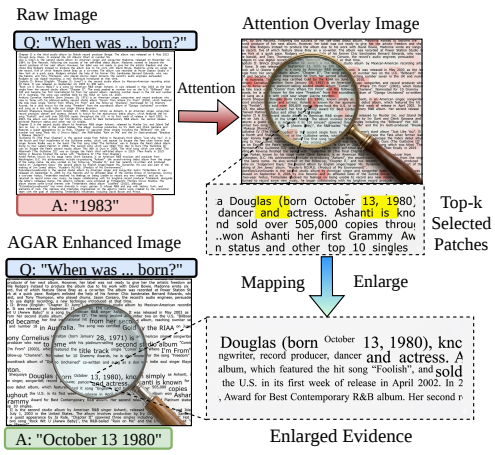


Figure 3: AGAR overview. Attention localizes evidence patches, which are magnified on a re-rendered page for a second forward pass.

Stage 2 – Magnified Re-rendering and Re-inference. We re-invoke the renderer with $\hat{\mathcal{E}}$ as the set of character spans to draw at a magnified font size s_{font}, $\mathcal{I}^{(1)} = \mathcal{R}(T, \hat{\mathcal{E}}, s_{\text{font}})$ (Alg. 1, line 5), while keeping all other typography and layout parameters identical to $\mathcal{I}^{(0)}$. We then produce the final answer with one more forward pass, $\hat{a} = \mathcal{M}(\mathcal{I}^{(1)}, q)$ (Alg. 1, line 6), and discard the first-pass output. This makes AGAR a model-agnostic plug-in for any VLM that exposes layer-wise attentions. No prompt or weight changes are needed.

5 Experiments

We evaluate AGAR around four questions: (i) does AGAR consistently improve VTC accuracy on off-the-shelf VLMs across short-form, long-context, and multi-page memory QA (§5.2)? (ii) does its gain persist when the underlying VLM is post-trained (§5.3)? (iii) is the gain robust to low-quality input images and irrelevant text contexts (§5.4)? and (iv) how sensitive is AGAR to its two core hyper-parameters, the patch budget K and the magnification factor s_{font} (§5.5)?

5.1 Experimental Setup

Datasets. We benchmark AGAR on three families of VTC tasks of increasing visual and textual complexity. (i) **Short-form QA**: NQ [13], HotpotQA [31] (distractor mode), TriviaQA [12] (TQA), and NewsQA [25], each rendered onto a single page. (ii) **LongBench [3] multi-page QA**: HotpotQA-LB (HQA), 2WikiMultihop [8] (2WMQA), MuSiQue [26] (MSQ), and Qasper [6] (QP), where the context spans multiple rendered pages. (iii) **LoCoMo [18] multi-page memory QA**, which further

stresses long, session-style contexts. We use the official validation/test splits and report EM and F1; per-dataset sample counts and average context lengths are summarized in Table 4 (Appendix B).

Models and Baselines. Our primary backbone is **Qwen3-VL-8B-Instruct**; cross-architecture results on **InternVL3.5-8B**, **GLM-4.1V-9B-Thinking**, and **Glyph** (post-trained on GLM-4.1V-9B-Thinking) are in Appendix C. We compare AGAR against three baseline groups: *Plain VQA* (answers directly from $\mathcal{I}^{(0)}$); two *naive baselines* that isolate AGAR's two design choices — *random*, which magnifies K randomly chosen patches (tests *which* patches matter), and *uniform*, which scales the whole page font to match AGAR's visual-token count (tests whether token count alone suffices); and prior emphasis methods adapted from VQA: *VEA* [17], *CGR*/*VAR* [15], and *AGLA* [1] (App. G).

Implementation. Pages render at 9 px; AGAR uses late-layer range $[a, b] = [0.5, 1.0]$ and font scale $s_{\text{font}} = 1.5$. The patch budget is $K = 8$ for single-page short-form QA and $K \approx 2\%$ of total patches for the multi-page benchmarks (LongBench, LoCoMo). See examples in App. I.

5.2 Main Results: AGAR on Off-the-Shelf VLMs

To answer question (i), we evaluate off-the-shelf Qwen3-VL-8B-Instruct on the abovementioned datasets. Short-form single-page QA results are reported in Tab. 2. Multi-page LongBench QA and LoCoMo memory QA results are reported jointly in Tab. 3. The two naive baselines (Random, Uniform) yield no or only marginal gains over Plain VQA. This indicates that AGAR's gain comes from effectively selecting the important evidence to emphasize, rather than simply increasing the number of visual tokens. The prior VQA-emphasis methods (VEA, CGR, VAR, AGLA) provide only limited enhancement and underperform our method, and even hurt the performance on some datasets. These methods are originally designed for standard VQA, where *de-emphasising* irrelevant visual background helps the answer. They thus import the wrong inductive bias for VTC, in which the surrounding text must remain readable for full context understanding. AGAR, in contrast, delivers clear and consistent gains over Plain VQA on every cell of both tables (e.g. $+18.1\%$ F1 on short-form HotpotQA, $+28.5\%$ on LB-MSQ multi-page, $+38.8\%$ on LoCoMo OpenDomain) and strictly outperforms every baseline, while increasing visual-token cost only marginally (maintaining $\sim 3\times$ compression). The same observation holds for the other three VLMs; see Appendix C, Tables 5–10. Appendix J further compares AGAR with the attention-guided cropping baseline ViCrop [35] and with delivering the selected spans as text, and reports confidence intervals and significance tests for the headline results.

Table 2: **Short-form single-page QA** on Qwen3-VL-8B. Each cell shows the metric followed by its *relative* change (%) vs. Plain VQA. Avg. text tokens ≈ 805 ($\sim 3\times$ compression).

	NQ		HotpotQA		NewsQA		TQA		Avg. tok.
Method	EM	F1	EM	F1	EM	F1	EM	F1	(↓)
Plain VQA	47.3	63.8	29.2	39.7	36.7	58.3	61.4	67.9	268
Random	47.1 (−0.4%)	63.7 (−0.2%)	30.4 (+4.1%)	41.8 (+5.3%)	37.6 (+2.5%)	58.7 (+0.7%)	63.2 (+2.9%)	70.4 (+3.7%)	301
Uniform	48.2 (+1.9%)	64.4 (+0.9%)	30.3 (+3.8%)	41.4 (+4.3%)	37.6 (+2.5%)	58.7 (+0.7%)	62.6 (+2.0%)	69.6 (+2.5%)	292
VEA-Br	47.2 (−0.2%)	63.9 (+0.2%)	30.0 (+2.7%)	40.5 (+2.0%)	36.2 (−1.4%)	57.7 (−1.0%)	60.7 (−1.1%)	68.6 (+1.0%)	268
VEA-Con	47.2 (−0.2%)	63.9 (+0.2%)	30.9 (+5.8%)	41.6 (+4.8%)	36.4 (−0.8%)	58.1 (−0.3%)	59.9 (−2.4%)	67.8 (−0.1%)	268
CGR	11.5 (−75.7%)	29.3 (−54.1%)	22.6 (−22.6%)	29.5 (−25.7%)	38.5 (+4.9%)	59.3 (+1.7%)	51.5 (−16.1%)	67.3 (−0.9%)	268
VAR	46.9 (−0.8%)	63.6 (−0.3%)	26.8 (−8.2%)	38.3 (−3.5%)	13.5 (−63.2%)	30.9 (−47.0%)	46.9 (−23.6%)	63.6 (−6.3%)	268
AGLA	48.8 (+3.2%)	64.6 (+1.3%)	30.2 (+3.4%)	41.2 (+3.8%)	37.2 (+1.4%)	58.7 (+0.7%)	63.2 (+2.9%)	70.2 (+3.4%)	268
AGAR (ours)	**50.2** (+6.1%)	**66.5** (+4.2%)	**34.3** (+17.5%)	**46.9** (+18.1%)	**39.0** (+6.3%)	**60.3** (+3.4%)	**64.7** (+5.4%)	**72.2** (+6.3%)	291

Takeaway 1. AGAR effectively improves VTC performance on off-the-shelf VLMs with only a marginal increase in visual-token cost.

5.3 Composing AGAR with Post-Training

Many real-world VTC applications [5, 23, 28] adapt VLMs via post-training, raising the question of whether AGAR's gain persists on post-trained VLMs. We test this on the short-form QA family (NQ / HotpotQA / NewsQA / TQA) under two settings. *(a) Qwen3-VL-8B-Instruct vs. its SFT counterpart* (Fig. 4(a)): we LoRA-fine-tune Qwen3-VL-8B-Instruct on *mix4*, a balanced 20k corpus sampled

Table 3: **Multi-page benchmarks** on Qwen3-VL-8B (F1). *Left*: LongBench multi-page QA (LB-HQA / LB-2WMQA / LB-MSQ / LB-QP). *Right*: LoCoMo multi-page memory QA (SingleHop / MultiHop / Temporal / OpenDomain). Each cell shows F1 followed by its *relative* change (%) vs. Plain VQA. Avg. text tokens $\approx$ 14.7k ($\sim 3\times$ compression).

	LongBench (multi-page) – F1				LoCoMo (multi-page memory) – F1				Avg. tok.
Method	**LB-HQA**	**LB-2WMQA**	**LB-MSQ**	**LB-QP**	**SingleHop**	**MultiHop**	**Temporal**	**OpenDom.**	(↓)
Plain VQA	39.8	37.5	17.2	29.5	26.1	20.5	22.7	12.1	4,916
Random	40.8 (+2.5%)	37.3 (−0.5%)	17.6 (+2.3%)	33.8 (+14.6%)	28.0 (+7.3%)	19.0 (−7.3%)	22.6 (−0.4%)	14.8 (+22.3%)	5,351
Uniform	39.8 (±0%)	37.5 (±0%)	16.9 (−1.7%)	33.3 (+12.9%)	26.5 (+1.5%)	18.5 (−9.8%)	21.8 (−4.0%)	12.2 (+0.8%)	5,378
VEA-Br	41.8 (+5.0%)	35.2 (−6.1%)	21.3 (+23.8%)	30.3 (+2.7%)	29.4 (+12.6%)	20.3 (−1.0%)	23.5 (+3.5%)	15.4 (+27.3%)	4,916
VEA-Con	42.1 (+5.8%)	36.1 (−3.7%)	18.2 (+5.8%)	29.9 (+1.4%)	27.8 (+6.5%)	19.2 (−6.3%)	23.1 (+1.8%)	13.5 (+11.6%)	4,916
CGR	34.1 (−14.3%)	36.6 (−2.4%)	21.2 (+23.3%)	29.3 (−0.7%)	30.1 (+15.3%)	21.8 (+6.3%)	21.2 (−6.6%)	15.8 (+30.6%)	4,916
VAR	37.7 (−5.3%)	39.2 (+4.5%)	18.5 (+7.6%)	30.9 (+4.7%)	24.3 (−6.9%)	17.1 (−16.6%)	20.0 (−11.9%)	16.0 (+32.2%)	4,916
AGLA	41.2 (+3.5%)	39.1 (+4.3%)	21.7 (+26.2%)	31.5 (+6.8%)	26.1 (±0%)	19.1 (−6.8%)	24.8 (+9.3%)	15.2 (+25.6%)	4,916
AGAR (ours)	**44.3** (+11.3%)	**40.8** (+8.8%)	**22.1** (+28.5%)	**34.2** (+15.9%)	**32.0** (+22.6%)	**21.9** (+6.8%)	**25.7** (+13.2%)	**16.8** (+38.8%)	5,384

from the four training splits (5k each, rendered at the standard 9 px, no attention-aware augmentation), and evaluate on the same held-out splits as §5.2. *(b) GLM-4.1V-9B-Thinking vs. Glyph* (Fig. 4(b)): **Glyph** [5] is a public third-party VTC model derived from GLM-4.1V-9B-Thinking via large-scale SFT+RL.

In both settings, AGAR is applied at inference only, with no change to training data, objective, or weights. As shown in Fig. 4, AGAR continues to lift F1 on top of every post-trained baseline across all four datasets in both regimes. On some datasets, the further gain AGAR delivers over the post-trained model is even comparable to, or exceeds, the gain post-training itself delivered: on *(a)* HotpotQA, Plain VQA $\rightarrow$ SFT lifts F1 by $+6.5$ absolute, and SFT $\rightarrow$ SFT + AGAR adds another $+5.5$; on *(b)* NQ, GLM-Thinking $\rightarrow$ Glyph lifts F1 by $+4.5$ while Glyph $\rightarrow$ Glyph + AGAR adds another $+8.5$ on top. AGAR therefore composes effectively with VLM post-training to deliver a further free inference-time boost.

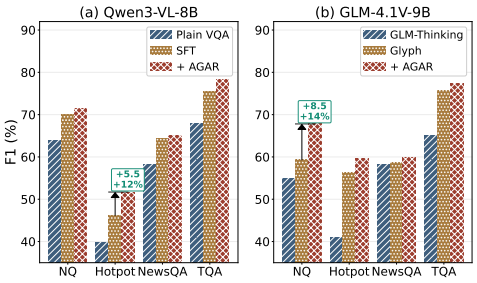


Figure 4: AGAR composes with post-training (short-form F1). **(a)** Qwen3-VL-8B vs. mix4 SFT. **(b)** GLM-4.1V-9B-Thinking vs. Glyph.

Takeaway 2. AGAR composes effectively with VLM post-training to further improve performance.

5.4 Robustness to Imperfect Inputs

We further ask whether AGAR remains useful under degraded inputs—common in real-world VTC where pages are scanned, compressed, or surrounded by retrieval noise. We probe two principal degradation sources on Qwen3-VL-8B: *(i) image-level perturbations* (low resolution, sensor noise, defocus blur) and *(ii) text-side distractors* surrounding the gold evidence (e.g., unrelated paragraphs from retrieval). Fig. 5 reports plain F1, AGAR F1, and the gain $\Delta = \mathrm{F1}_{\mathrm{enh}} - \mathrm{F1}_{\mathrm{plain}}$.

Visual perturbations. We perturb the rendered HotpotQA pages (1000 samples) with three families of image corruption applied directly to the PNG before inference: resolution reduction (downsample by $\{25\%, 50\%, 75\%\}$, which simultaneously shrinks the visual-token budget from 332 to $\{185, 79, 22\}$ tokens), additive Gaussian noise on pixel intensities ($\sigma \in \{0.2, 0.4, 0.6\}$ on the $[0, 1]$ range), and Gaussian blur ($\sigma \in \{1, 2, 3\}$ pixels); plain vs. AGAR inference is then run on the perturbed images (Fig. 5 (a)–(c)). The main observation is that AGAR is robust to visual degradation and recovers a large fraction of the F1 that Plain VQA loses under it. Under resolution reduction, for example, AGAR at 79 visual tokens (50% downsample) matches Plain VQA at 185 tokens (25% downsample) — AGAR effectively buys back a full level of image quality / token budget. Moreover, in some regimes AGAR's recovery actually *grows* with the severity of the perturbation: blur σ=2 px yields $\Delta = +17.6$ (+65%) and Gaussian noise σ=0.6 yields $\Delta = +11.4$ (+48%), against an unperturbed baseline of $\Delta = +6.8$ (+16%).

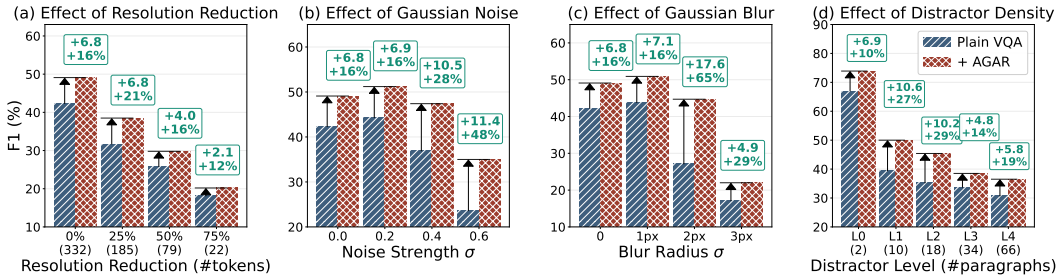


Figure 5: **AGAR robustness on Qwen3-VL-8B / HotpotQA.** Blue: Plain F1; orange: AGAR F1; box: absolute and relative Δ. **(a)–(c)**: **visual** corruptions (downsampling, Gaussian noise, blur). **(d)**: **text-side** dilution from L0 (gold only) to L4 (66 paragraphs, with hard negatives at L2–L4).

Text-side irrelevance. We build a 5-level distractor-density ladder on HotpotQA (1000 samples per level) mimicking a retrieval pipeline of increasing noise: L0 = 2 gold only; L1 = native HotpotQA distractor (2 gold + 8 distractors); L2–L4 add $\{8, 24, 56\}$ hard negatives drawn from other HotpotQA training questions. Gold paragraphs are always present but randomly positioned. Plain F1 cliffs at L0→L1 ($67.0 \rightarrow 39.4$, -27.6) and then drifts down only mildly ($\rightarrow 30.7$ at L4). AGAR's Δ stays positive throughout, with the relative gain at every distractor level exceeding the clean-input (L0) baseline, and peaks under moderate clutter (L1: $+10.6$, $+27\%$; L2: $+10.2$, $+29\%$).

Takeaway 3. AGAR is robust to both visual and text-side input degradation, and effectively recovers much of the F1 loss caused by quality degradation.

5.5 Hyperparameter Analysis

To answer (iv), we study AGAR's sensitivity to its two hyperparameters: the fraction k of patches to magnify and the font scale α applied to the covered words. We sweep $k \in \{0.5\%, 1\%, 2\%, 5\%, 10\%\}$ and $\alpha \in \{1.2, 1.5, 1.8, 2.0\}$ on all datasets from §5.2. Fig. 6 shows HotpotQA (single-page) and LB-QP (multi-page) as representatives; the full sweep is in Appendix A. AGAR is *robust* to (k, α): nearly all cells exceed the Plain VQA plane (HotpotQA: $20/20$, LB-QP: $19/20$; $11/12$ appendix subtasks have $\geq 90\%$ above-plane cells). Small (k, α) underperform due to insufficient magnification against attention dilution; larger values trade tokens for accuracy, tunable to budget. Well-chosen pairs yield pronounced gains (HotpotQA F1 $+14.3$/$+35.9\%$ at $(10\%, 2.0)$). Other hyperparameters (base font size, layer range) are discussed in App. H and F.

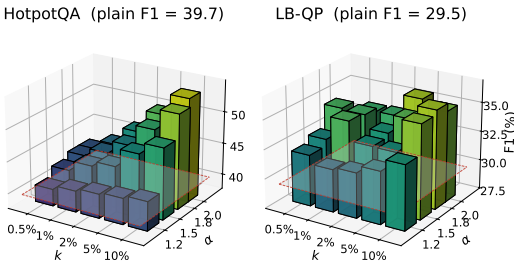


Figure 6: **AGAR's** (k, α) **landscape** on Qwen3-VL-8B on HotpotQA and LB-QP. The red dashed plane marks Plain VQA.

Takeaway 4. AGAR is robust to its (k, α) hyperparameters: most pairs already improve over Plain VQA, while stronger (k, α) trade additional visual tokens for substantially larger gains.

6 Conclusion and Limitation

We studied how VLMs internally process Visual Text Comprehension (VTC) inputs and where their failures come from. VLMs reliably localize evidence in middle-to-late layers, yet this localization is decoupled from answer correctness, leaving them in a *localization-without-utilization* regime. Magnifying the localized evidence already substantially repairs these failures. Motivated by this, we propose **AGAR**, a plug-and-play method that uses the model's own attention to enlarge identified evidence patches and re-query the same VLM. Across four open-weight VLMs and 12 subtasks, AGAR consistently improves over Plain VQA, outperforms training-free baselines, composes with post-training, and stays robust under visual- and text-side degradation.

Limitations. AGAR requires access to the backbone's attention scores, limiting its use on closed-source VLMs. Deeper mechanistic accounts (e.g., circuit-level analysis) are left to future work.

Acknowledgments and Disclosure of Funding

Shenglai Zeng, Kai Guo, Xinnan Dai, Xianxuan Long, and Hui Liu are supported by the National Science Foundation (NSF) under grant numbers IIS2212032, IIS2212144, IIS2504089, DUE2234015, CNS2246050, and DRL2405483, the Michigan Department of Agriculture and Rural Development, and the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under Award Number DE-SC0026773.

References

[1] Wenbin An, Feng Tian, Sicong Leng, Jiahao Nie, Haonan Lin, QianYing Wang, Ping Chen, Xiaoqin Zhang, and Shijian Lu. Mitigating object hallucinations in large vision-language models with assembly of global and local attention. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29915–29926, 2025.

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[3] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3119–3137, 2024.

[4] Gabriela Ben Melech Stan, Estelle Aflalo, Raanan Yehezkel Rohekar, Anahita Bhiwandiwalla, Shao-Yen Tseng, Matthew Lyle Olson, Yaniv Gurwicz, Chenfei Wu, Nan Duan, and Vasudev Lal. Lvlm-intrepret: An interpretability tool for large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8182–8187, 2024.

[5] Jiale Cheng, Yusen Liu, Xinyu Zhang, Yulin Fei, Wenyi Hong, Ruiliang Lyu, Weihan Wang, Zhe Su, Xiaotao Gu, Xiao Liu, et al. Glyph: Scaling context windows via visual-text compression. *arXiv preprint arXiv:2510.17800*, 2025.

[6] Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, 2021.

[7] Lang Feng, Fuchao Yang, Feng Chen, Xin Cheng, Haiyang Xu, Zhenglin Wan, Ming Yan, and Bo An. Agentocr: Reimagining agent history via optical self-compression. *arXiv preprint arXiv:2601.04786*, 2026.

[8] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, 2020.

[9] Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

[10] Jukka Hyönä and Robert F Lorch. Effects of topic headings on text processing: Evidence from adult readers' eye fixation patterns. *Learning and instruction*, 14(2):131–152, 2004.

[11] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[12] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, 2017.

[13] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

[14] Qing'an Liu, Juntong Feng, Yuhao Wang, Xinzhe Han, Yujie Cheng, Yue Zhu, Haiwen Diao, Yunzhi Zhuge, and Huchuan Lu. Vista-bench: Do vision-language models really understand visualized text as well as pure text? *arXiv preprint arXiv:2602.04802*, 2026.

[15] Yexin Liu, Zhengyang Liang, Yueze Wang, Xianfeng Wu, Feilong Tang, Muyang He, Jian Li, Zheng Liu, Harry Yang, Sernam Lim, et al. Unveiling the ignorance of mllms: Seeing clearly, answering incorrectly. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9087–9097, 2025.

[16] Zhining Liu, Rana Ali Amjad, Ravinarayana Adkathimar, Tianxin Wei, and Hanghang Tong. Selfelicit: Your language model secretly knows where is the relevant evidence. *arXiv preprint arXiv:2502.08767*, 2025.

[17] Zhining Liu, Ziyi Chen, Hui Liu, Chen Luo, Xianfeng Tang, Suhang Wang, Joy Zeng, Zhenwei Dai, Zhan Shi, Tianxin Wei, et al. Seeing but not believing: Probing the disconnect between visual attention and answer correctness in vlms. *arXiv preprint arXiv:2510.17771*, 2025.

[18] Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870, 2024.

[19] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[20] Shutong Qiao, Wei Yuan, Tong Chen, Xiangyu Zhao, Quoc Viet Hung Nguyen, and Hongzhi Yin. When text-as-vision meets semantic ids in generative recommendation: An empirical study. *arXiv preprint arXiv:2601.14697*, 2026.

[21] Keith Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372, 1998.

[22] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[23] Yaorui Shi, Shugui Liu, Yu Yang, Wenyu Mao, Yuxin Chen, Qi Gu, Hui Su, Xunliang Cai, Xiang Wang, and An Zhang. Memocr: Layout-aware visual memory for efficient long-horizon reasoning. *arXiv preprint arXiv:2601.21468*, 2026.

[24] Qwen Team. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

[25] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, 2017.

[26] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

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

[28] Yibo Wang, Yongcheng Jing, Shunyu Liu, Hao Guan, Rong-cheng Tu, Chengyu Wang, Jun Huang, and Dacheng Tao. Vtc-r1: Vision-text compression for efficient long-context reasoning. *arXiv preprint arXiv:2601.22069*, 2026.

[29] Haoran Wei, Yaofeng Sun, and Yukun Li. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*, 2025.

[30] Haoran Wei, Yaofeng Sun, and Yukun Li. Deepseek-ocr 2: Visual causal flow. *arXiv preprint arXiv:2601.20552*, 2026.

[31] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380, 2018.

[32] Shenglai Zeng, Pengfei He, Kai Guo, Tianqi Zheng, Hanqing Lu, Yue Xing, and Hui Liu. Towards context-robust llms: A gated representation fine-tuning approach. *arXiv preprint arXiv:2502.14100*, 2025.

[33] Shenglai Zeng, Jiankun Zhang, Bingheng Li, Yuping Lin, Tianqi Zheng, Dante Everaert, Hanqing Lu, Hui Liu, Yue Xing, Monica Xiao Cheng, et al. Towards knowledge checking in retrieval-augmented generation: A representation perspective. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2952–2969, 2025.

[34] Shenglai Zeng, Tianqi Zheng, Chuan Tian, Dante Everaert, Yau-Shian Wang, Yupin Huang, Michael J Morais, Rohit Patki, Jinjin Tian, Xinnan Dai, et al. Attn-gs: Attention-guided context compression for efficient personalized llms. *arXiv preprint arXiv:2602.07778*, 2026.

[35] Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. MLLMs know where to look: Training-free perception of small visual details with multimodal LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025.

[36] Hongbo Zhao, Meng Wang, Fei Zhu, Wenzhuo Liu, Bolin Ni, Fanhu Zeng, Gaofeng Meng, and Zhaoxiang Zhang. Vtcbench: Can vision-language models understand long context with vision-text compression? *arXiv preprint arXiv:2512.15649*, 2025.

[37] Jianping Zhong, Guochang Li, Chen Zhi, Junxiao Han, Zhen Qin, Xinkui Zhao, Nan Wang, Shuiguang Deng, and Jianwei Yin. Can vision-language models handle long-context code? an empirical study on visual compression. *arXiv preprint arXiv:2602.00746*, 2026.

APPENDIX

(§A) Full (k, α) Sensitivity Sweep 13
(§B) Dataset Statistics 14
(§C) Cross-Architecture Results 14
(§D) Layer-wise Evidence Attention on Additional Datasets 14
(§E) Per-head NDCG Heatmaps on Additional Datasets 15
(§F) Layer-range Aggregated NDCG 15
(§G) Baseline Implementation 15
(§H) Impact on Base Token Font 16
(§I) Case Study 17
(§J) Additional Baselines and Statistical Significance 25
(§K) Experiments Compute Resources 26
(§L) Broader Impact 26
(§M) Prompt Used by AGAR 26

A Full (k, α) Sensitivity Sweep

Fig. 7 extends the headline 1×2 view of §5.5 (HotpotQA + LB-QP) to all 12 subtasks used in §5.2. Each panel reports enhanced F1 (%) at every $(k, \alpha) \in \{0.5\%, 1\%, 2\%, 5\%, 10\%\} \times \{1.2, 1.5, 1.8, 2.0\}$ cell; the translucent grey plane (red dashed wireframe) marks the Plain VQA baseline, and per-panel colour is normalised to the panel's own range.

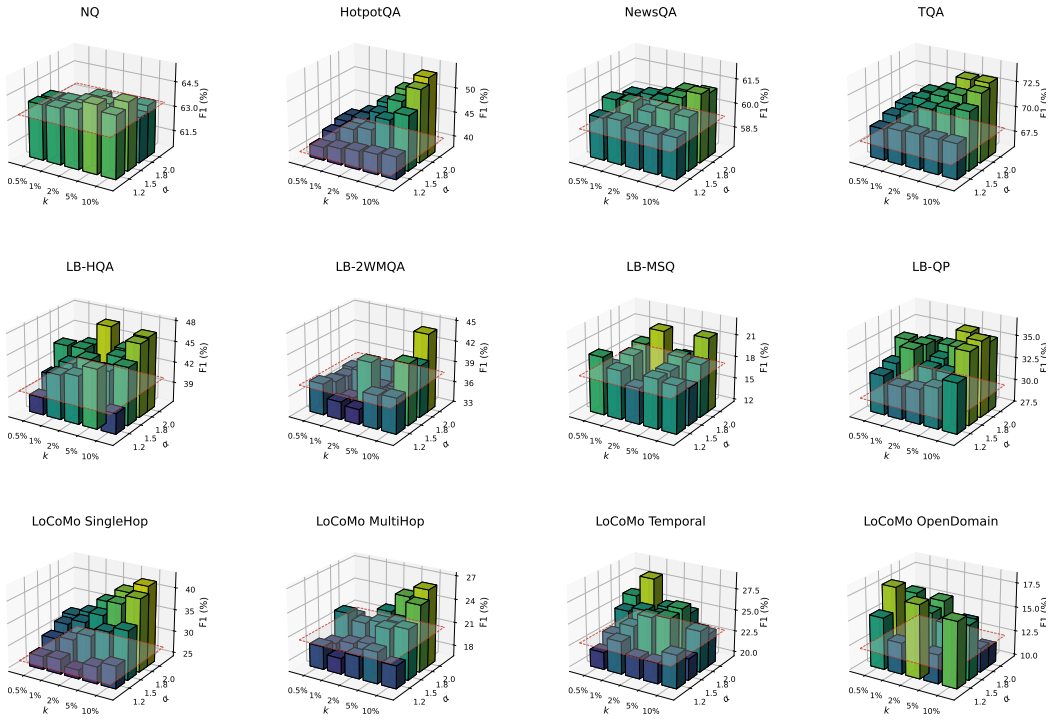


Figure 7: **Full (k, α) sweep on 12 subtasks.** 3D bar version. Short-form QA (top row): NQ / HotpotQA / NewsQA / TQA. LongBench multi-page (middle row): LB-HQA / LB-2WMQA / LB-MSQ / LB-QP. LoCoMo multi-page memory (bottom row): SingleHop / MultiHop / Temporal / OpenDomain.

B Dataset Statistics

Table 4 summarizes the three benchmark families used in §5: short-form QA (single rendered page), LongBench multi-page QA, and LoCoMo multi-page memory QA. Numbers are computed on the official validation/test split of each dataset (HotpotQA in distractor mode); context length is reported in both raw characters and whitespace-separated words.

Table 4: Dataset statistics for the three benchmark families used in §5. **Avg. chars** and **Avg. words** are the per-sample mean character and whitespace-separated word counts of the textual context T before rendering. All datasets are used consistently with their licenses for non-commercial academic research.

Family	Dataset	# Samples	Avg. chars	Avg. words	License
Short-form QA (single page)	NQ	12,836	843	159	CC BY-SA 3.0
	HotpotQA	7,404	5,479	897	CC BY-SA 4.0
	NewsQA	4,212	3,001	493	MIT
	TQA	7,785	4,064	673	Apache 2.0
LongBench (multi-page)	LB-HQA	200	56,550	9,133	MIT
	LB-2WMQA	200	29,615	4,873	MIT
	LB-MSQ	200	69,405	11,196	MIT
	LB-QP	200	23,641	3,599	MIT
Memory QA	LoCoMo	1,542	79,668	14,394	CC BY-NC 4.0

C Cross-Architecture Results

To verify that the AGAR gain is not specific to one architecture or post-training recipe, we additionally evaluate three other VLMs: **InternVL3.5-8B**, **GLM-4.1V-9B-Thinking**, and **Glyph** (a public model post-trained from GLM-4.1V-9B-Thinking specifically for visual text comprehension). Tables 5–10 mirror Tables 2–3 of the main text. The magnitude and direction of the gain track the same patterns observed for Qwen3-VL-8B in the main text.

Table 5: **InternVL3.5-8B short-form QA** (F1).

Method	NQ	HotpotQA	NewsQA	TQA
Plain VQA	59.1	31.5	44.4	66.5
Random	59.3 (+0.3%)	31.6 (+0.3%)	44.6 (+0.5%)	66.4 (−0.2%)
Uniform	59.6 (+0.8%)	31.9 (+1.3%)	44.9 (+1.1%)	67.1 (+0.9%)
VEA-Br	59.0 (−0.2%)	29.4 (−6.7%)	40.8 (−8.1%)	62.8 (−5.6%)
VEA-Con	59.9 (+1.4%)	29.6 (−6.0%)	40.3 (−9.2%)	65.3 (−1.8%)
CGR	33.6 (−43.1%)	19.8 (−37.2%)	29.0 (−34.7%)	35.0 (−47.4%)
VAR	52.4 (−11.3%)	25.9 (−17.7%)	42.6 (−4.1%)	59.5 (−10.5%)
AGLA	53.0 (−10.3%)	30.2 (−4.1%)	40.6 (−8.6%)	61.7 (−7.2%)
AGAR (ours)	**67.4** (+14.0%)	**32.8** (+4.1%)	**47.2** (+6.3%)	**68.4** (+2.9%)

D Layer-wise Evidence Attention on Additional Datasets

Fig. 8 extends the HotpotQA panel of Fig. 1 to NQ / TriviaQA / NewsQA across the same four VLMs (Qwen3-VL-8B, InternVL3.5-8B, GLM-4.1V-9B-Thinking, and Glyph). Each cell shows per-layer relative attention to Evidence vs. Non-Evidence image tokens, with samples split by answer correctness (Correct / Incorrect) and aggregated as mean±1 std. The flat-then-rising profile observed on HotpotQA — Evidence attention is roughly equal to Non-Evidence in shallow layers and pulls sharply ahead from the late-middle layers onward — replicates across all three additional datasets and all four models.

Table 6: **InternVL3.5-8B multi-page benchmarks** (F1).

Method	LongBench (multi-page) – F1				LoCoMo (multi-page memory) – F1			
	LB-HQA	**LB-2WMQA**	**LB-MSQ**	**LB-QP**	**SingleHop**	**MultiHop**	**Temporal**	**OpenDom.**
Plain VQA	48.6	43.1	24.0	37.2	35.9	26.0	28.1	16.4
Random	47.5 (−2.3%)	42.9 (−0.5%)	23.8 (−0.8%)	37.3 (+0.3%)	38.6 (+7.5%)	26.9 (+3.5%)	28.3 (+0.7%)	17.2 (+4.9%)
Uniform	48.6 (±0%)	43.1 (±0%)	24.0 (±0%)	37.2 (±0%)	35.8 (−0.3%)	26.0 (±0%)	28.1 (±0%)	16.4 (±0%)
VEA-Br	48.3 (−0.6%)	31.3 (−27.4%)	7.1 (−70.4%)	21.0 (−43.5%)	15.3 (−57.4%)	13.1 (−49.6%)	31.4 (+11.7%)	8.1 (−50.6%)
VEA-Con	37.0 (−23.9%)	32.9 (−23.7%)	10.1 (−57.9%)	22.4 (−39.8%)	17.2 (−52.1%)	15.0 (−42.3%)	31.1 (+10.7%)	8.1 (−50.6%)
CGR	24.9 (−48.8%)	23.8 (−44.8%)	9.5 (−60.3%)	19.5 (−47.7%)	12.2 (−66.2%)	10.3 (−60.4%)	5.7 (−79.6%)	9.9 (−39.4%)
VAR	30.0 (−38.2%)	30.2 (−30.0%)	13.4 (−44.0%)	12.4 (−66.7%)	11.6 (−67.8%)	13.5 (−47.9%)	15.6 (−44.4%)	11.8 (−28.0%)
AGLA	31.7 (−34.8%)	29.5 (−31.5%)	17.8 (−25.7%)	17.9 (−51.9%)	15.3 (−57.5%)	14.6 (−44.0%)	16.6 (−41.1%)	14.9 (−9.5%)
AGAR (ours)	**48.9** (+0.6%)	**43.8** (+1.6%)	**25.6** (+6.7%)	**38.0** (+2.2%)	**42.2** (+17.5%)	**27.6** (+6.2%)	**31.4** (+11.7%)	**17.8** (+8.5%)

Table 7: **GLM-4.1V-9B-Thinking short-form QA** (F1).

Method	NQ	HotpotQA	NewsQA	TQA
Plain VQA	54.8	40.9	58.1	65.1
Random	54.7 (−0.2%)	41.0 (+0.2%)	57.9 (−0.3%)	65.3 (+0.3%)
Uniform	55.4 (+1.1%)	41.4 (+1.2%)	58.4 (+0.5%)	65.7 (+0.9%)
VEA-Br	43.9 (−19.9%)	39.2 (−4.2%)	58.0 (−0.2%)	66.4 (+2.0%)
VEA-Con	47.3 (−13.7%)	40.7 (−0.5%)	58.4 (+0.5%)	65.2 (+0.2%)
CGR	47.4 (−13.5%)	36.9 (−9.8%)	49.7 (−14.5%)	67.4 (+3.5%)
VAR	58.6 (+7.0%)	36.9 (−9.8%)	57.3 (−1.4%)	67.6 (+3.8%)
AGLA	55.9 (+2.0%)	43.2 (+5.6%)	49.5 (−14.8%)	67.2 (+3.2%)
AGAR (ours)	**61.8** (+12.8%)	**48.0** (+17.4%)	**58.5** (+0.7%)	**68.1** (+4.6%)

E Per-head NDCG Heatmaps on Additional Datasets

Fig. 9 extends the HotpotQA panel of Fig. 2 to NQ / TriviaQA / NewsQA. Each cell shows the per-head NDCG of last-token attention to evidence tokens (layer × head), with the colour scale centred on the per-cell median NDCG so that white maps to median and red/blue mark heads above or below it. The same population of late-middle–layer evidence-attending heads observed on HotpotQA is visible across all three additional datasets and all four VLMs.

F Layer-range Aggregated NDCG

Table 11 reports the NDCG of last-token attention to evidence patches after pooling head-averaged attention over different ranges of decoder layers: the first half of layers (*Early*, 0–50% depth), the second half (*Late*, 50–100%), and the full stack (*All*, 0–100%). Across every (model, dataset) cell, the late-half pooling yields the highest NDCG, confirming that evidence-localizing attention is concentrated in the second half of the network and that aggregating the early half dilutes rather than reinforces the signal.

G Baseline Implementation

CGR operates in two stages. First, the model is prompted with the rendered image and question to extract relevant content as structured textual evidence (verbatim snippets, names, dates, numbers, and brief surrounding context). Second, this extracted text is concatenated with the original question and passed to a text-only stage that produces the final answer, abstaining when evidence is insufficient.

VAR constructs a binary visual mask from the model's final-layer attention scores. We aggregate attention across heads, select the globally highest-scoring patches, and present the model with a version of the image in which those patches are highlighted and the remainder is unchanged.

AGLA is run using the official codebase and default hyperparameters. The method produces a question-conditioned saliency map, masks out low-relevance regions to form a local-evidence image, and ensembles generations from the original and masked images via AGLA decoding. No AGLA-specific parameters were tuned on our evaluation sets.

Table 8: **GLM-4.1V-9B-Thinking multi-page benchmarks** (F1).

	LongBench (multi-page) – F1				LoCoMo (multi-page memory) – F1			
Method	**LB-HQA**	**LB-2WMQA**	**LB-MSQ**	**LB-QP**	**SingleHop**	**MultiHop**	**Temporal**	**OpenDom.**
Plain VQA	44.5	50.1	24.8	24.8	29.2	17.8	11.9	13.7
Random	43.5 (−2.2%)	46.5 (−7.2%)	25.5 (+2.8%)	23.9 (−3.6%)	28.6 (−2.1%)	20.2 (+13.5%)	11.5 (−3.4%)	9.4 (−31.4%)
Uniform	44.5 (±0%)	50.2 (+0.2%)	24.8 (±0%)	24.2 (−2.4%)	29.2 (±0%)	17.8 (±0%)	11.9 (±0%)	13.7 (±0%)
VEA-Br	46.7 (+4.9%)	50.2 (+0.2%)	24.7 (−0.4%)	23.6 (−4.8%)	31.0 (+6.2%)	17.8 (±0%)	12.6 (+5.9%)	10.5 (−23.4%)
VEA-Con	45.9 (+3.1%)	49.9 (−0.4%)	23.9 (−3.6%)	24.9 (+0.4%)	29.8 (+2.1%)	18.3 (+2.8%)	12.1 (+1.7%)	13.3 (−2.9%)
CGR	45.8 (+2.9%)	24.7 (−50.8%)	21.9 (−11.7%)	19.9 (−19.7%)	27.5 (−6.0%)	14.8 (−16.8%)	4.9 (−59.1%)	7.2 (−47.2%)
VAR	45.6 (+2.6%)	50.3 (+0.4%)	24.5 (−1.2%)	22.7 (−8.4%)	32.5 (+11.2%)	19.4 (+8.9%)	11.8 (−0.7%)	11.7 (−14.4%)
AGLA	45.7 (+2.7%)	38.9 (−22.4%)	26.5 (+6.8%)	18.6 (−25.0%)	32.2 (+10.3%)	17.4 (−2.2%)	9.0 (−24.4%)	7.7 (−43.8%)
AGAR (ours)	**46.9** (+5.4%)	**50.7** (+1.2%)	**26.8** (+8.1%)	**25.9** (+4.4%)	**32.7** (+12.0%)	**22.0** (+23.6%)	**13.5** (+13.4%)	**16.2** (+18.2%)

Table 9: **Glyph short-form QA** (F1).

Method	**NQ**	**HotpotQA**	**NewsQA**	**TQA**
Plain VQA	59.3	56.4	57.1	75.8
Random	63.0 (+6.2%)	56.9 (+0.9%)	55.3 (−3.2%)	74.4 (−1.8%)
Uniform	62.0 (+4.6%)	53.4 (−5.3%)	56.3 (−1.4%)	73.4 (−3.2%)
VEA-Br	49.4 (−16.7%)	51.0 (−9.6%)	54.8 (−4.0%)	70.2 (−7.4%)
VEA-Con	50.5 (−14.8%)	50.5 (−10.5%)	54.5 (−4.6%)	71.0 (−6.3%)
CGR	56.2 (−5.2%)	57.1 (+1.2%)	47.0 (−17.6%)	76.1 (+0.4%)
VAR	66.9 (+12.8%)	52.9 (−6.2%)	55.3 (−3.2%)	75.9 (+0.1%)
AGLA	62.0 (+4.6%)	49.2 (−12.7%)	49.6 (−13.2%)	75.7 (−0.1%)
AGAR (ours)	**67.8** (+14.3%)	**59.6** (+5.7%)	**60.1** (+5.3%)	**77.3** (+2.0%)

VEA-Br / VEA-Con are the brightness and contrast variants of Visual Evidence Augmentation [17]. We follow the default hyperparameters described in the paper (Appendix A.3, "VEA and Baselines Implementation Details"): attention is aggregated by layer-span mean over the middle-to-late half ($[0.5, 1.0]$), with σ=0.1, α=0.9, offset = 1.0. VEA-Br up-weights pixel brightness in attended regions while VEA-Con increases local contrast; in both cases the augmented image replaces the plain page for a single forward pass.

H Impact on Base Token Font

This section studies the impact of the base token font on AGAR. We evaluate Qwen3-VL-8B-Instruct on HotpotQA and LB-QP under two degraded renderings: 5× compression (5 px base font size) and 10× compression (3 px base font size). Figures 11 and 12 report the results under two enlargement rules. Both variants select the top 2% of attention patches and differ only in the enlargement rule:

- FIXED uses a fixed target: selected regions are redrawn at 13.5 px no matter how small the compressed base font size is. This spends more visual tokens but explicitly targets a readable font size.
- RELATIVE uses a relative target: selected regions are enlarged by 1.5× from their current base font size, yielding 40.5 px for the original 27 px rendering, 7.5 px under 5× compression, and 4.5 px under 10× compression. This keeps the compressed-setting token overhead very small.

The fixed-size figure also includes a 3× compression panel (9 px base font size; HotpotQA n=7404, LB-QP n=200), where the fixed 13.5 px target is exactly a 1.5× enlargement of the selected 9 px regions. The relative-size figure includes the original 27 px rendering as a readable reference point for the same relative rule.

Takeaway. Across all compression ratios and both enlargement rules, AGAR consistently improves F1 over compressed PLAINVQA.

Table 10: **Glyph multi-page benchmarks** (F1).

	LongBench (multi-page) – F1				LoCoMo (multi-page memory) – F1			
Method	**LB-HQA**	**LB-2WMQA**	**LB-MSQ**	**LB-QP**	**SingleHop**	**MultiHop**	**Temporal**	**OpenDom.**
Plain VQA	61.9	68.2	45.7	38.4	42.0	29.1	21.2	23.0
Random	61.7 (−0.3%)	70.3 (+3.1%)	50.9 (+11.4%)	37.2 (−3.1%)	43.0 (+2.4%)	26.9 (−7.6%)	19.0 (−10.4%)	20.9 (−9.1%)
Uniform	62.0 (+0.2%)	66.1 (−3.1%)	45.7 (±0%)	36.2 (−5.7%)	42.0 (±0%)	29.1 (±0%)	21.2 (±0%)	23.1 (+0.4%)
VEA-Br	11.6 (−81.3%)	65.9 (−3.4%)	41.6 (−9.0%)	37.4 (−2.6%)	41.4 (−1.4%)	27.2 (−6.5%)	19.8 (−6.6%)	18.5 (−19.6%)
VEA-Con	60.4 (−2.4%)	66.3 (−2.8%)	43.0 (−5.9%)	38.0 (−1.0%)	43.0 (+2.4%)	30.4 (+4.5%)	22.2 (+4.7%)	18.2 (−20.9%)
CGR	61.0 (−1.5%)	52.1 (−23.7%)	50.8 (+11.2%)	39.0 (+1.6%)	36.5 (−13.1%)	27.3 (−6.2%)	15.7 (−26.0%)	14.6 (−36.6%)
VAR	60.4 (−2.4%)	69.1 (+1.3%)	45.4 (−0.6%)	38.4 (±0%)	43.2 (+2.9%)	29.4 (+1.0%)	22.0 (+3.8%)	21.5 (−6.4%)
AGLA	56.6 (−8.5%)	50.0 (−26.7%)	44.0 (−3.7%)	31.4 (−18.1%)	38.5 (−8.4%)	28.0 (−3.8%)	21.9 (+3.3%)	18.6 (−19.0%)
AGAR (ours)	**63.6** (+2.7%)	**72.5** (+6.3%)	**52.2** (+14.2%)	**40.4** (+5.2%)	**44.5** (+6.0%)	**30.7** (+5.5%)	**23.2** (+9.4%)	**24.3** (+5.7%)

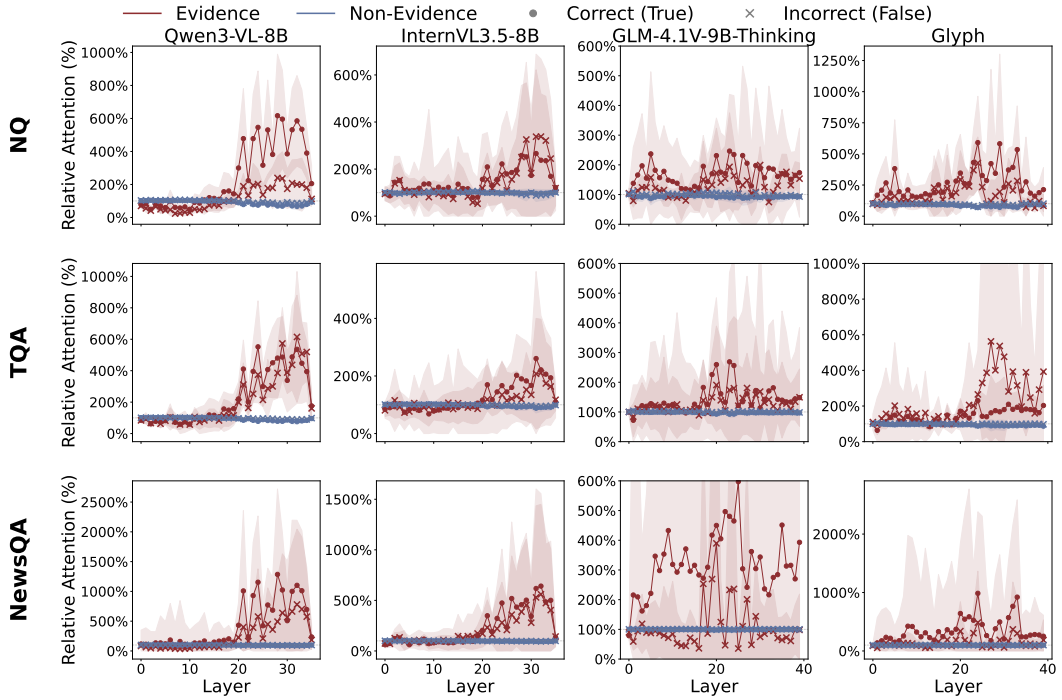


Figure 8: **Layer-wise relative attention on NQ / TriviaQA / NewsQA (appendix companion to Fig. 1)**.

I Case Study

We provide case studies on various datasets from Figure 13 to Figure 22.

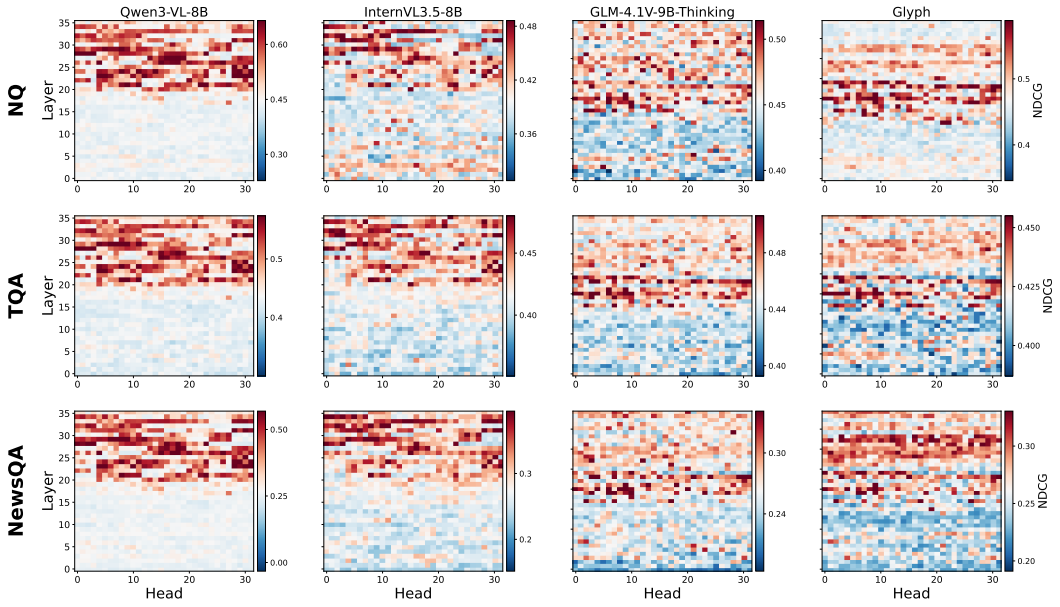


Figure 9: **Per-head NDCG of last-token attention to evidence tokens on NQ / TriviaQA / NewsQA (appendix companion to Fig. 2)**.

Table 11: **Layer-range aggregated NDCG** (mean over n=100 samples per cell). Each cell is the NDCG of last-token attention, head-averaged and then pooled over the specified depth range. The Late (50–100%) pool dominates in every (model, dataset) cell.

Model	**Dataset**	**Early** (0–50%)	**Late** (50–100%)	**All** (0–100%)
Qwen3-VL-8B	NQ	0.439	**0.606**	0.535
	TriviaQA	0.428	**0.540**	0.507
	NewsQA	0.264	**0.469**	0.396
	HotpotQA	0.634	**0.709**	0.689
InternVL3.5-8B	NQ	0.392	**0.454**	0.435
	TriviaQA	0.402	**0.466**	0.440
	NewsQA	0.264	**0.412**	0.359
	HotpotQA	0.609	**0.643**	0.629
GLM-4.1V-9B-Thinking	NQ	0.455	**0.468**	0.460
	TriviaQA	0.417	**0.434**	0.424
	NewsQA	0.263	**0.290**	0.274
	HotpotQA	0.610	**0.636**	0.620
Glyph	NQ	0.497	**0.555**	0.520
	TriviaQA	0.415	**0.441**	0.422
	NewsQA	0.260	**0.317**	0.277
	HotpotQA	0.612	**0.645**	0.621

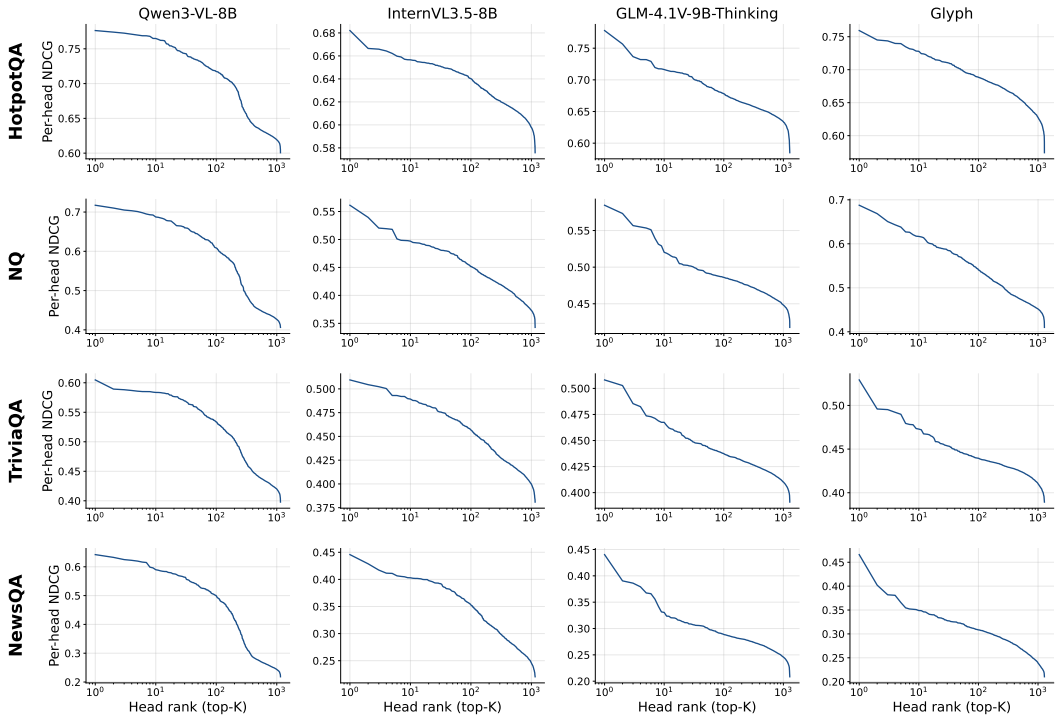


Figure 10: **Per-head NDCG vs. head rank** for the same four VLMs $\times$ four datasets as Fig. 9. Each curve sorts the (layer, head) heads by mean NDCG (descending) on a log-scale rank axis; the steep drop near the top and slow decay toward the median quantify the head-population structure visible in the layer $\times$ head heatmaps.

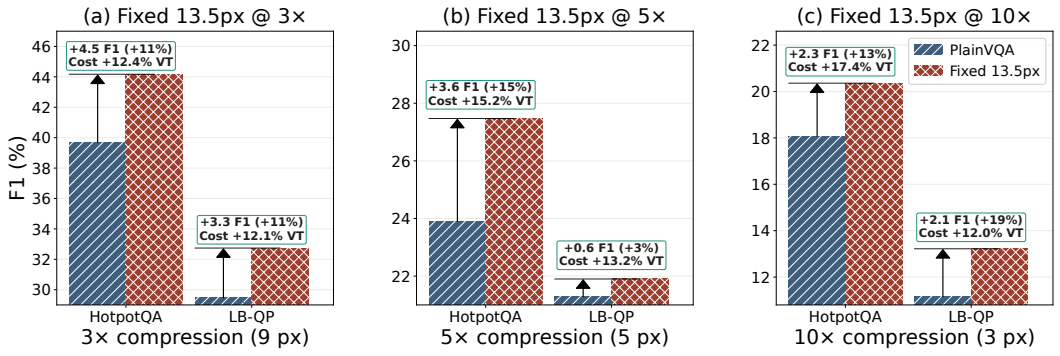


Figure 11: **Fixed readable-size enlargement on Qwen3-VL-8B under vision-token compression.** Blue: PLAINVQA F1; red: FIXED F1. Panel (a) compares the Qwen PLAINVQA baseline at $3\times$ compression (9 px) with FIXED under the same rendering, where fixed 13.5 px equals scale $1.5\times$; panels (b) and (c) show FIXED at $5\times$ compression (5 px) and $10\times$ compression (3 px). Boxes report absolute and relative ΔF1 plus vision-token cost.

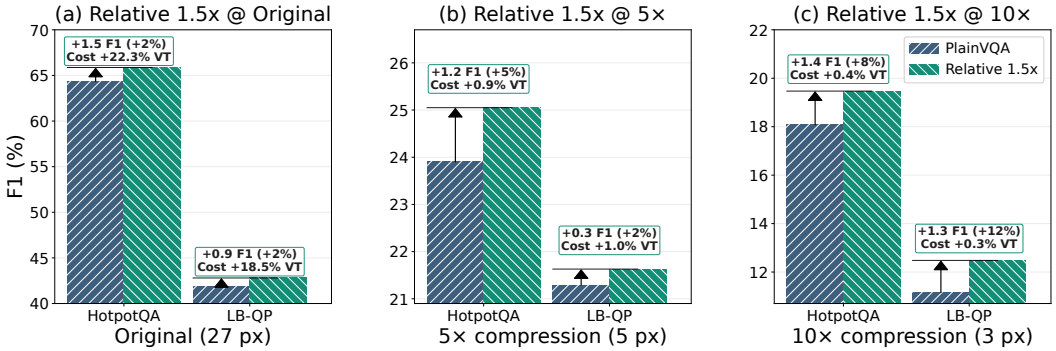


Figure 12: **Relative-scale enlargement on Qwen3-VL-8B under vision-token compression.** Blue: PLAINVQA F1; green: RELATIVE F1. Panel (a) shows RELATIVE on the original 27 px rendering; panels (b) and (c) show RELATIVE at 5× compression (5 px) and 10× compression (3 px). Boxes report absolute and relative ΔF1 plus vision-token cost.



Figure 13: LoCoMo case for James's 2021 trip. **Evidence:** in a March 2022 dialogue, James says he visited Italy last year, while Japan is John's later trip. The enhanced answer changes Japan to Italy.

Question: What kind of professional experience did Gina get accepted for on May 23, 2023?

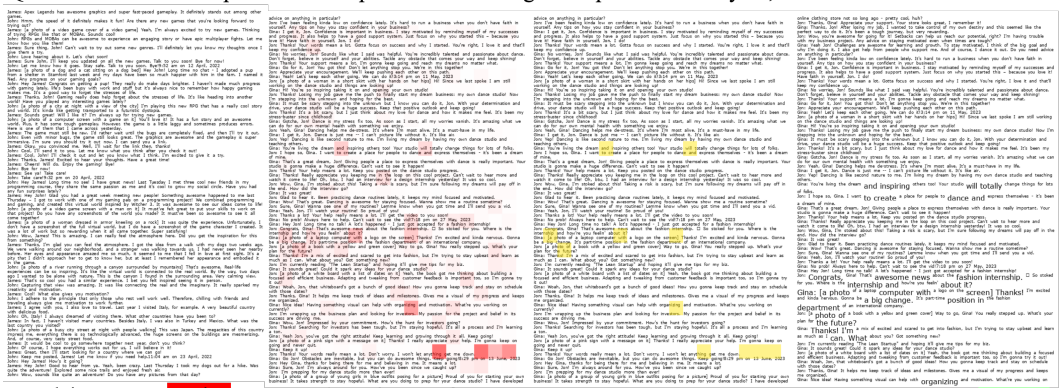


Figure 14: LoCoMo case for Gina's accepted professional experience. **Evidence:** Gina says she got accepted for a fashion internship; the design internship is only an earlier interview. The enhanced answer changes design internship to fashion internship.

Question: What book did Caroline recommend to Melanie?

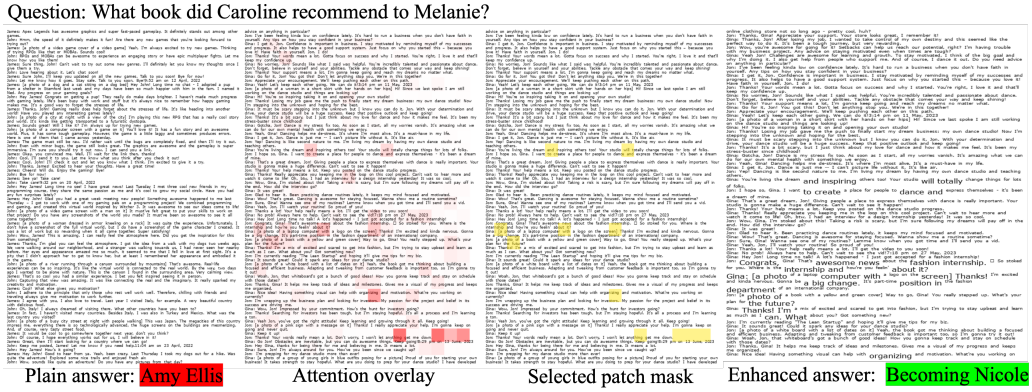
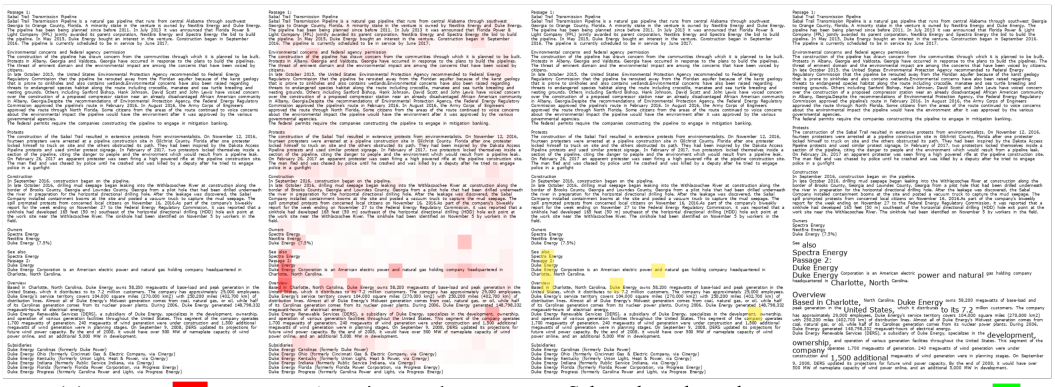


Figure 15: LoCoMo case for Caroline's book recommendation. **Evidence:** Caroline says she loved *Becoming Nicole* by Amy Ellis Nutt and highly recommends it. The enhanced answer corrects the title to *Becoming Nicole*.

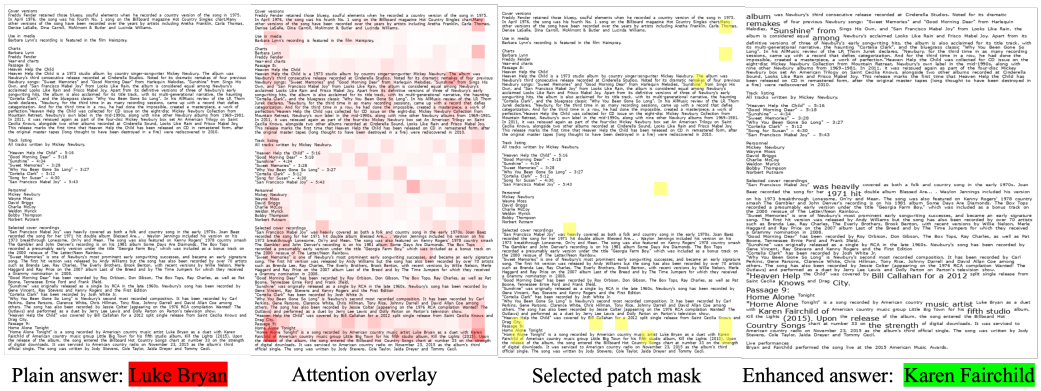
Question: Are both Duke Energy and Affiliated Managers Group based in Massachusetts?



Plain answer: Yes | Attention overlay | Selected patch mask | Enhanced answer: No

Figure 16: LongBench HotpotQA case asking whether Duke Energy and Affiliated Managers Group are both based in Massachusetts. **Evidence:** Duke Energy is headquartered in Charlotte, North Carolina, so the conjunction is false even though Affiliated Managers Group has a Massachusetts office. The enhanced answer changes Yes to No.

Question: Who sings Home Alone Tonight with the Rain is a Good Thing singer?



Plain answer: Luke Bryan | Attention overlay | Selected patch mask | Enhanced answer: Karen Fairchild

Figure 17: LongBench MuSiQue case linking *Rain Is a Good Thing* to *Home Alone Tonight*. **Evidence:** *Rain Is a Good Thing* is by Luke Bryan, and *Home Alone Tonight* is Luke Bryan's duet with Karen Fairchild. The enhanced answer changes Luke Bryan to Karen Fairchild.



Figure 18: LongBench NarrativeQA case on Anthony Rogers's 1927 business. **Evidence:** the story says his company was the American Corporation. The enhanced answer gives the requested business type, radioactive gas, instead of the broader name.

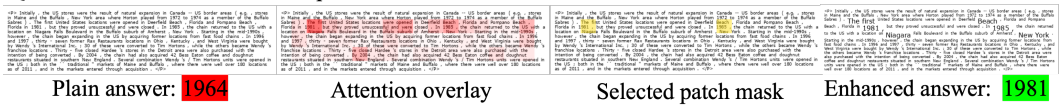


Figure 19: Natural Questions case on when Tim Hortons opened in the USA. **Evidence:** the first United States locations opened in Deerfield Beach and Pompano Beach, Florida, in 1981. The enhanced answer changes 1964 to the U.S. opening year, 1981.



Figure 20: Natural Questions case on which apostle spoke at the Council. **Evidence:** Barnabas and Paul report their ministry, but the apostle James quotes Amos and submits the accepted proposal. The enhanced answer changes Paul to James.

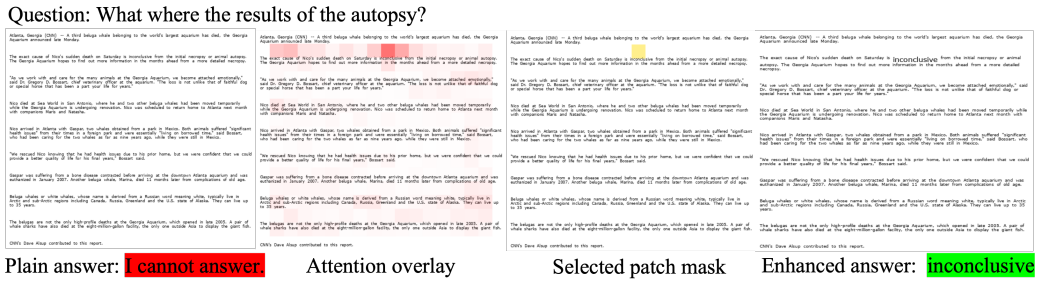


Figure 21: NewsQA case on the autopsy result. **Evidence:** the initial necropsy, or animal autopsy, found Nico's cause of death inconclusive. The enhanced answer changes the plain no-answer response to inconclusive.

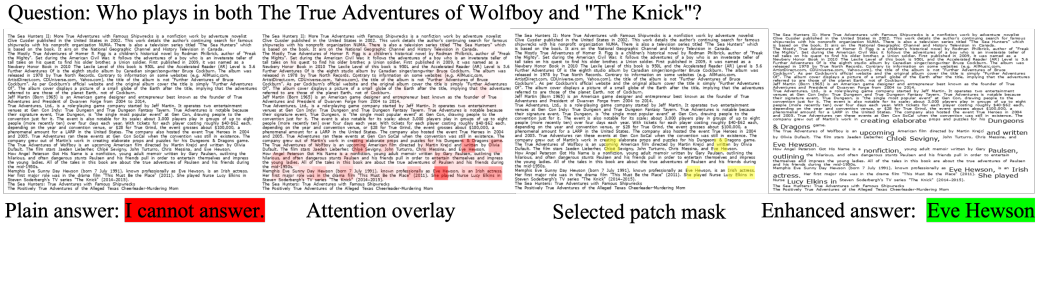


Figure 22: HotpotQA case on the actor shared by *The True Adventures of Wolfboy* and *The Knick*. **Evidence:** *Wolfboy* stars Eve Hewson, and Eve Hewson played Nurse Lucy Elkins in *The Knick*. The enhanced answer recovers Eve Hewson from the plain no-answer response.

J Additional Baselines and Statistical Significance

This section complements the main results in §5 with (i) a comparison against an attention-guided two-pass cropping baseline, (ii) a text-side baseline that delivers the first-pass selection as text rather than by re-rendering, and (iii) confidence intervals and significance tests for the headline results. All experiments use Qwen3-VL-8B-Instruct with the same pipeline and hyper-parameters as the main text.

Attention-guided cropping (ViCrop). ViCrop [35] is a training-free two-pass method for natural-image VQA that reads layer-averaged relative attention, crops the most-attended region, and feeds the up-scaled crop alongside the original image. We reproduce it with its generic instruction and layer averaging, with sliding-window sizes matching its absolute crop sizes; the crop is up-scaled to the full-page token budget and kept alongside the original page, and on multi-page inputs it is placed on the highest-importance page. Table 12 reports F1 and total visual tokens on n=1000 samples per short-form dataset (n=200 for LB-QP). AGAR outperforms ViCrop on all five datasets while adding only $+3$–9% visual tokens over Plain VQA, against ViCrop's $+80$–120%; on NQ, ViCrop even falls below Plain VQA. Crop-and-upscale duplicates a region at the cost of the page's token budget, whereas AGAR reallocates the budget within a single page without deleting or occluding the surrounding context.

Table 12: Comparison with ViCrop [35] on Qwen3-VL-8B. Each cell shows F1 followed by the average number of visual tokens per sample (@). n=1000 per short-form dataset, n=200 for LB-QP.

Dataset	Plain VQA	ViCrop	AGAR (ours)
NQ	64.2 @ 90	63.7 @ 163	**66.2** @ 97
NewsQA	57.8 @ 419	59.8 @ 857	**60.1** @ 426
TQA	67.6 @ 233	69.0 @ 489	**72.0** @ 271
HotpotQA	40.2 @ 305	43.0 @ 687	**46.8** @ 331
LB-QP (multi-page)	30.4 @ 1506	31.5 @ 1940	**34.4** @ 1611

Text-side delivery of the selected spans. To test whether the gain comes from evidence *selection* alone or also requires in-page magnification, we keep AGAR's first-pass selection fixed and deliver it as text instead of re-rendering. *Words-only* feeds the selected words alone in place of the page; *Words-appended* concatenates them to the plain page. Table 13 averages over the four short-form datasets (n=1000 each) and over the eight multi-page LongBench and LoCoMo test sets (900 samples). Words-only falls far below Plain VQA at both scales, since extracting the evidence removes the surrounding context the model needs to reason. Words-appended improves slightly over Plain VQA, confirming that the selected words are question-relevant, but remains well below AGAR, and the gap widens on multi-page inputs. A correct selection therefore still has to be delivered through re-rendering.

Table 13: Delivering the first-pass selection as text (F1, Qwen3-VL-8B). Short-form averages NQ, HotpotQA, NewsQA, and TQA; multi-page averages the eight LongBench and LoCoMo test sets.

Setting	Plain VQA	Words-only	Words-appended	AGAR (ours)
Short-form (avg.)	57.5	50.1	58.3	**60.8**
Multi-page (avg.)	27.3	20.2	28.7	**32.0**

Statistical significance. Table 14 reports paired bootstrap 95% confidence intervals (10,000 resamples over per-sample records) for the short-form F1 results in Tab. 2. All four gains are significant under an exact McNemar test ($p < 10^{-4}$). On the multi-page suite in Tab. 3, all eight cells are positive, with a pooled paired difference of $+4.3$ F1 (95% CI $[+3.0, +5.6]$, $p \leq 10^{-4}$).

Table 14: Paired bootstrap 95% confidence intervals for short-form F1 on Qwen3-VL-8B. Δ is the paired per-sample difference between AGAR and Plain VQA.

	NQ	**HotpotQA**	**NewsQA**	**TQA**
Plain VQA	63.8 [63.1, 64.5]	39.7 [38.8, 40.8]	58.3 [57.0, 59.5]	67.9 [67.6, 69.5]
AGAR (ours)	66.5 [65.8, 67.2]	46.9 [45.9, 47.9]	60.3 [59.0, 61.5]	72.2 [71.2, 73.1]
Δ (paired)	+2.7 [+2.2, +3.2]	+7.0 [+6.0, +8.0]	+2.0 [+1.3, +2.7]	+3.6 [+2.8, +4.4]

K Experiments Compute Resources

All experiments are run on Python 3.10 with CUDA 12.8 and PyTorch in `bfloat16`. AGAR is inference-only and does not require training, so the dominant cost is two forward passes per question (one to read attention, one on the re-rendered page). A single GPU with at least 50 GB of free memory is recommended (e.g. NVIDIA A100-80G, H100, or H200); on shorter inputs a 40 GB card is sufficient, while the multi-page LongBench / LoCoMo settings benefit from the larger memory to fit the full visual-token sequence. Our experiments were carried out on a $2\times$ A100-80G server.

L Broader Impact

AGAR is a training-free, plug-and-play enhancement on top of an existing vision–language model: it does not collect new data, does not fine-tune weights, and does not change the prompt. Its effect is to redistribute visual capacity on a re-rendered page so that the model's own attention decides which words appear at a more legible scale before the second forward pass. We discuss potential broader impacts at a high level below, and we keep the discussion deliberately broad rather than tied to specific deployments.

Potential positive impacts. By making vision-text comprehension more reliable on long, dense, or visually compressed pages, AGAR can plausibly contribute to more useful document, screenshot, and memory QA assistants under tight context-window or token budgets, including settings where readers benefit from larger, more salient text (e.g. accessibility-oriented or low-bandwidth scenarios). Because the method exposes which patches the model itself attended to, the resulting magnified renderings also make the model's internal evidence selection more inspectable, which may support debugging, auditing, and downstream interpretability research on how VLMs read visualized text.

Potential negative impacts. AGAR inherits all general risks of VLM-based question answering: hallucinated answers, biased or unsafe outputs, leakage of sensitive content present in the input page, and uneven reliability across languages, domains, scripts, and user populations. Because the method is built on the model's own attention, any systematic bias in what the underlying VLM tends to attend to will likewise be magnified on the re-rendered page, which could amplify rather than mitigate spurious focus on certain regions. Like most accuracy-oriented post-processing techniques, AGAR could also be misused as a component in larger pipelines for surveillance, automated content scraping, or other applications whose downstream effects depend on context, deployment, and policy choices outside the scope of this paper. We therefore view AGAR as a research artifact: it should be evaluated together with appropriate human oversight and use-case-specific safety considerations before being deployed in any high-stakes setting.

M Prompt Used by AGAR

AGAR uses a single short instruction template across all four backbones (Qwen3-VL-8B, InternVL3.5-8B, GLM-4.1V-9B-Thinking, Glyph) and across both the first (attention-reading) and second (re-rendered) forward passes; the same template is also used by Plain VQA, so that AGAR's gain over Plain VQA reflects the input image alone. The template asks for a direct answer grounded in the visible text on the rendered page and includes an explicit abstention option, and is shown in Table 15.

Table 15: Prompt template used by AGAR (and Plain VQA). `{question}` is replaced by the dataset's natural-language question; nothing else in the prompt is altered between models or datasets.

```
Directly answer the question based on the text shown in the image, no
explanation is needed.  If the image does not contain any evidence, output
“I cannot answer based on the given context.”
Question:  {question}
```

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 accurately reflects the content in the intro, method, and experiments

   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: See Conclusion and Limitation Section.

   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: No theory included

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 fully disclose all the information needed to reproduce the main experimental results of the paper. We provide anonymous GitHub repos.

   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: Code and instructions to reproduce the main results are publicly available at `https://github.com/AGAR-NeurIPS2026/AGAR-NeurIPS2026`.

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: Please check Section Experiment for details

   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: Please check preliminaries and our experiments.

   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: Check Experiment Section and Appendix.

   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: Yes.

   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: [Yes]

    Justification: See Appendix and Conclusion.

    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: No risk.

    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: See Appendix.

    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 do not release new datasets or pretrained models—all experiments use existing public benchmarks (NQ, HotpotQA, NewsQA, TriviaQA, LongBench, LoCoMo) under their original licenses. We do release the code for AGAR at an anonymous URL provided in the abstract; the repository includes a README with installation, usage, and reproduction instructions.

    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: Not applicable.

    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: 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: [Yes]

    Justification: We use (V)LLM's internal signal to design our methods, see Section 4.

    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.