HiToc: Hierarchical Token Compression Across the Vision-Language Pipeline for Long Document Question Answering

Anonymous authors
Paper under double-blind review

Abstract

Long-document question answering is computationally expensive due to long visual-token sequences. Compressing this context without disrupting document structure or losing cross-page evidence remains challenging. We propose *HiToc*, a training-free hierarchical token compression framework that progressively refines visual context across the vision-language pipeline. After visual encoding, *Bottom-up Background Token Merging* reduces background redundancy while preserving layout cues. During prefilling, *Adaptive Page-level Token Assignment* converts question-conditioned attention into soft page budgets and then selects tokens within each page, reducing attention-sink effects and preserving complementary cross-page evidence. During generation, *layer-adaptive dynamic KV-cache compression* reselects active visual context across decoder depths at each step. Across four document QA benchmarks, HiToc reduces the final active visual-token budget to under 20% of the original input while preserving 94.94–98.72% of baseline performance and achieving $1.50\times$–$4.05\times$ LLM-stage speedups.

1 Introduction

Vision-language models (VLMs) increasingly support document understanding directly from page images, retaining access to text, tables, figures, and their spatial organization (Ye et al., 2023a;b; Bai et al., 2025; Qwen Team, 2025). Extending these capabilities to long documents, however, introduces a substantial inference burden: documents with dozens of pages can produce thousands of visual tokens, although only a small fraction is typically relevant to a given question. Efficient long-document QA therefore requires substantial compression without discarding the evidence and structural cues needed for correct answering.

The central difficulty is that *visual redundancy is not always a reliable proxy for answer relevance*. A blank region may still convey layout clues, evidence may be distributed across pages, and useful visual context can vary across decoder depths and generation steps. Compression decisions must therefore account for both *what signals are available to assess importance* and *what context may still be needed later in inference*.

Existing methods provide useful primitives for token merging, attention-based ranking, and KV-cache compression (Bolya et al., 2023; Chen et al., 2024; Li et al., 2024). Recent document-specific methods also exploit background redundancy and question-dependent sparsity (Zhang et al., 2024; Choi et al., 2026) for token compression. However, long-document QA exposes a particular tension in how these operations are applied. Aggressive visual compression before question–document interaction risks removing content before its relevance can be assessed. Global token ranking by cross-modal attention remains vulnerable to attention sinks and competition across pages. Hard selection at the page level may eliminate an entire source of evidence due to a page-ranking error. Finally, selecting a fixed visual subset based only on the input question ignores the fact that useful evidence can vary across different stages of answer generation.

We address this challenge with **HiToc**, a training-free **Hi**erarchical **To**ken **C**ompression framework across the vision-language pipeline. The key idea is to *progressively refine the visual context using the information available at each stage of inference*. The hierarchy follows the computation itself: preserve document structure after visual encoding, allocate question-relevant evidence during

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

prefilling, and adapt the active visual context during decoding. Each stage passes a smaller visual context to the next while relying on increasingly informative signals for selection.

First, *Bottom-up Background Token Merging* (BBTM) performs conservative compression after the visual encoder. Lightweight text-aware patch labeling identifies candidate background regions, whose encoded tokens are merged and then reinserted according to their original positions. This reduces background redundancy while preserving useful layout cues and foreground content for later question-conditioned processing. Since label generation runs alongside visual encoding, the exposed overhead is limited to the lightweight merging operation.

Second, *Adaptive Page-level Token Assignment* (APTA) operates during LLM prefilling, when question–document interaction provides a stronger relevance signal. APTA first allocates soft token budgets across pages and then selects tokens within each page. This hierarchical page-to-token allocation makes token selection less vulnerable to attention sinks: tokens are not forced to compete through a single global ranking, and pages that are not top-ranked can still contribute evidence. As a result, APTA better preserves complementary information across pages.

Third, *layer-adaptive dynamic KV-cache compression* further refines the active visual context during generation. As decoding proceeds, the usefulness of visual tokens varies with both the generated context and decoder depth. We therefore divide the decoder into two stages and reselect the active visual KV entries at each stage based on the current decoding context, allowing different decoder depths to access different subsets of relevant visual evidence while further reducing computation.

Experiments on public multi-page DocVQA benchmarks show that HiToc reduces the final active visual-token budget to 3.6–18.2% of the original input while preserving 94.94–98.72% of baseline performance, with more than $1.5\times$ LLM-stage speedup. Our contributions are threefold:

- We propose **HiToc**, a training-free hierarchical visual token compression framework for long-document QA that progressively refines visual context across the VLM inference pipeline, from visual encoding to prefilling and decoding.
- We design stage-specific compression mechanisms aligned with stage-wise signals: layout-preserving background token merging, question-conditioned page-to-token allocation to reduce attention sinks and preserve cross-page evidence, and layer-adaptive visual KV selection during decoding.
- Extensive experiments on four multi-page DocVQA benchmarks demonstrate that HiToc delivers strong efficiency gains with minimal performance loss.

2 Related Work

2.1 Document understanding and document-specific compression

Recent multimodal document understanding models extend VLMs to text-rich and multi-page inputs (Ye et al., 2023a;b; Feng et al., 2024; Liao et al., 2025; Nacson et al., 2025), while long-document benchmarks further require cross-page evidence aggregation and reasoning over complex layouts (Tito et al., 2022; Ma et al., 2024; Chia et al., 2024). Existing document-specific compression methods remain limited for long-document QA: TokenCorr (Zhang et al., 2024) exploits patch correlations but is not question-conditioned, and DocPrune (Choi et al., 2026) prunes large portion of tokens before QA interaction. In contrast, HiToc performs progressive compression inside the QA backbone, preserving layout cues after visual encoding and using question-aware signals later to retain cross-page evidence.

2.2 Visual token compression in VLMs

Prior work studies token pruning and merging for efficient vision transformers and VLMs (Rao et al., 2021; Liang et al., 2022; Bolya et al., 2023; Chen et al., 2024; Zhang et al., 2025b; Alvar et al., 2025; Yang et al., 2025; Shang et al., 2025; He et al., 2024; Xing et al., 2024; Zhang et al., 2025a; Ding et al., 2026). HiToc builds on these ideas but adapts them to document QA: it uses *layout-aware background merging* instead of generic token merging, and replaces global token ranking with *question-conditioned page-to-token allocation*, which better preserves structural cues and complementary evidence across pages.

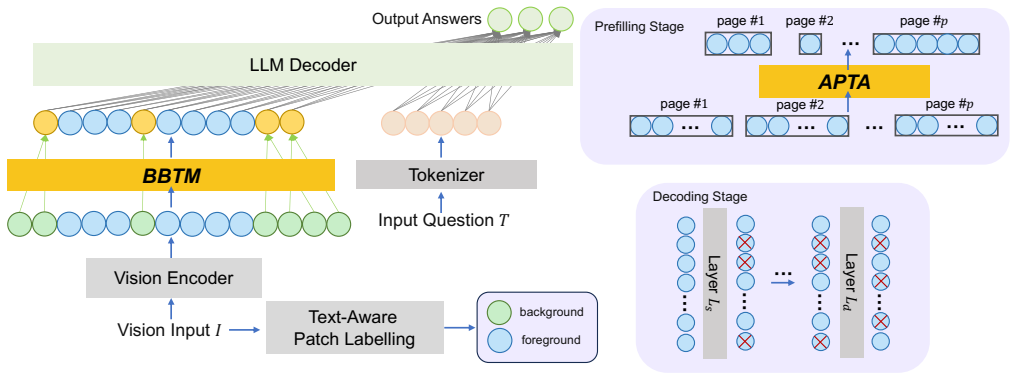

Figure 1: The overall structure of our proposed HiToc framework.

2.3 KV-CACHE COMPRESSION

Recent KV-cache compression methods improve decoding efficiency using attention statistics, layer depth, or temporal dynamics (Zhang et al., 2023; Li et al., 2024; Cai et al., 2025; Zeng et al., 2026). HiToc extends this idea to document QA by making visual KV access both *stage-aware* and *reactivable*: instead of fixing a single visual subset, it reselects active visual evidence across decoder stages during generation, allowing different depths to access different document regions when evidence is spatially distributed.

3 METHODS

In this section, we firstly present the overall architecture of our proposed HiToc model in Fig. 1 and then explicitly introduce the three submodules contained in HiToc.

3.1 BOTTOM-UP BACKGROUND TOKEN MERGING

Document images typically contain substantial background regions that generate redundant tokens but still convey layout cues. BBTM therefore performs text-aware patch labeling followed by background token merging, shortening the visual sequence before it enters the LLM while preserving foreground content and spatial organization.

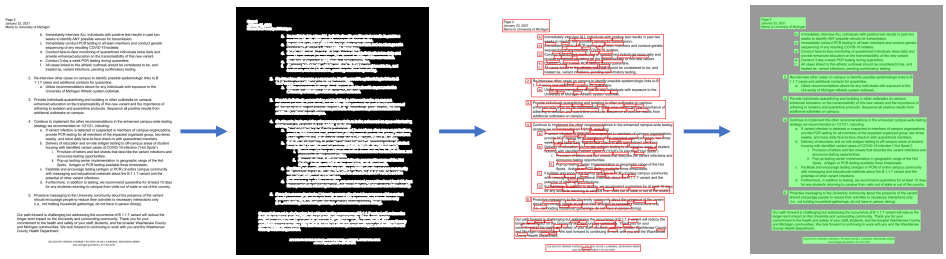

Figure 2: Text-aware patch labeling process in BBTM.

3.1.1 TEXT-AWARE PATCH LABELING

Given an input document image $I \in \mathbb{R}^{H\times W\times 3}$, we assign a text-aware label to each patch on the vision encoder's grid. We first apply Sobel edge detection (Sobel & Feldman, 1990):

$$E = \sqrt{G_x^2 + G_y^2}, \tag{1}$$

where G_x and G_y are the horizontal and vertical image gradients. We binarize E and apply morphological dilation to connect fragmented text strokes. Connected components above a minimum area threshold are retained as candidate text regions, each of which produces a binary mask $M_k \in \{0,1\}^{H\times W}$. All masks are aggregated into the saliency map $S = \sum_{k=1}^{K} M_k$. Fig. 2 illustrates this process and more implementation details are provided in Appendix C.1.

To align S with the visual encoder's patch grid, we partition it into non-overlapping patches at the corresponding stride (e.g., 32×32). For a patch $P_{i,j}$ at grid position (i, j), its text-space ratio is

$$r_{i,j} = \frac{\sum_{(x,y)\in P_{i,j}} \mathbf{1}[S(x,y) > 0]}{|P_{i,j}|}, \tag{2}$$

where $|P_{i,j}|$ is the number of pixels in the patch. Its binary label is then defined as:

$$l_{i,j} = \mathbf{1}[r_{i,j} > \tau]. \tag{3}$$

Patches with $l_{i,j} = 1$ are classified as text patches, while those with $l_{i,j} = 0$ are treated as background ones and subjected to the merging process described below.

3.1.2 Background Token Merging

Let $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ denote the ordered visual-token sequence produced by the vision encoder, with corresponding labels $\mathcal{L} = \{l_1, l_2, \ldots, l_N\}$ obtained by flattening the patch grid in token order. We preserve all text tokens ($l_i = 1$) without modification. For each contiguous background token sequence ($l_i = 0$) within a page, we apply the following rule according to its length n:

- **Single token ($n = 1$):** Preserve the token as-is.
- **Two tokens ($n = 2$):** Merge the two tokens by averaging their features.
- **Multiple tokens ($n > 2$):** We apply bipartite soft matching (Bolya et al., 2023) to reduce the group to $\lceil n/2 \rceil$ tokens. Specifically, tokens are split by alternating positions into source and destination sets. Each source is matched to its most similar destination by cosine similarity, and the top-$\lfloor n/2 \rfloor$ matches are merged by averaging each destination with its matched sources; unmatched tokens are preserved.

To maintain spatial consistency, the same grouping is applied to the corresponding 3D position coordinates (t, h, w). The merged background tokens and preserved foreground tokens are then reassembled according to their updated position indices, yielding the compressed visual sequence $\mathcal{V}'$.

3.2 Adaptive Page-level Token Assignment

While BBTM reduces spatial redundancy, retained content still varies in relevance across pages. Direct global selection from question-guided attention is vulnerable to attention sinks (Xiao et al., 2024) and inter-page competition. APTA therefore allocates token budgets across pages before selecting tokens within each page, allowing non-top-ranked pages to retain complementary evidence.

3.2.1 Page-level Relevance Estimation

At layer L_p during LLM prefilling, we compute the partial cross-modal attention from question-token sequence $\mathcal{T}$ to the visual sequence $\mathcal{V}'$ output by BBTM. For each attention head,

$$\mathbf{A}^{(h)} = \mathrm{softmax}\left(\frac{\mathbf{Q}_{\mathcal{T}}^{(h)}(\mathbf{K}_{\mathcal{V}'}^{(h)})^{\top}}{\sqrt{d_h}}\right) \in \mathbb{R}^{|\mathcal{T}|\times|\mathcal{V}'|}, \tag{4}$$

where $\mathbf{Q}_{\mathcal{T}}^{(h)} \in \mathbb{R}^{|\mathcal{T}|\times d_h}$, $\mathbf{K}_{\mathcal{V}'}^{(h)} \in \mathbb{R}^{|\mathcal{V}'|\times d_h}$ are the query and key projections, d_h is the head dimension.

For an input containing P pages, $\mathcal{V}'$ is segmented into page groups $\{\mathcal{V}'_1, \ldots, \mathcal{V}'_P\}$. We estimate the relevance score s_p for each page p by aggregating the attention weights allocated to its tokens:

$$s_p = \sum_{j:\, v_j \in \mathcal{V}'_p} \bar{a}_j, \qquad \bar{a}_j = \frac{1}{H_a|\mathcal{T}|}\sum_{h=1}^{H_a}\sum_{i=1}^{|\mathcal{T}|} a_{i,j}^{(h)}. \tag{5}$$

Here $a_{i,j}^{(h)}$ denotes the (i,j)-th entry of $\mathbf{A}^{(h)}$, H_a is the number of attention heads. The page retention ratio is then derived by re-normalizing the relevance scores relative to the most attended page:

$$\rho_p = \frac{s_p}{\max_{p'} s_{p'}}. \tag{6}$$

The most attended page will retain all its tokens, while other pages are compressed proportionally to their relative scores.

3.2.2 Intra-Page Token Selection

Given ρ_p, the token budget for page p is calculated as $k_p = \lceil |\mathcal{V}'_p| \cdot \rho_p \rceil$. We retain its k_p highest-scoring visual tokens:

$$\hat{\mathcal{V}}_p = \text{Top-}k_p \left(\mathcal{V}'_p; \{\bar{a}_j : v_j \in \mathcal{V}'_p\} \right). \tag{7}$$

The operator returns visual tokens ranked by $\bar{a}_j$ and the final sequence $\hat{\mathcal{V}}$ concatenates the selected tokens while preserving their original within-page and page order.

With page-specific budgets, a token competes within its page rather than against every token in the document. Soft allocation also avoids discarding an entire page solely because it is not top-ranked. This reduces the adverse effects of attention sinks and helps preserve complementary cross-page evidence when page ranking is imperfect.

3.3 Layer-Adaptive Dynamic KV-Cache Compression

BBTM compresses visual tokens before the LLM, and APTA selects question-relevant tokens during prefilling. During generation, however, the critical visual tokens vary with both decoding context and layer depth. We therefore divide the decoder into shallow and deep stages and dynamically select visual KV entries for each stage.

Given an LLM decoder with L layers, the shallow stage spans layers $L_s, \ldots, L_d - 1$, and the deep stage spans layers $L_d, \ldots, L$, where $L_s < L_d$. At the start layer of each stage $s \in \{\text{shallow}, \text{deep}\}$, let q_t denote the current query vector at decoding step t, and let $\mathcal{K}_{vis} = \{\mathcal{K}_1, \ldots, \mathcal{K}_{|\hat{\mathcal{V}}|}\}$ denote the keys corresponding to the visual tokens retained after BBTM and APTA. We compute their attention scores and retain the top-$k_d\%$ visual key-value pairs with the highest attention scores. Only these pairs are activated through the layers in the current stage s, others are evicted temporarily from the active cache, reducing KV access and computational cost during the decoding stage.

4 Experiments

4.1 Experimental Settings

Benchmarks and metrics. We evaluate on DUDE (Van Landeghem et al., 2023), MMDA (Xu et al., 2026), MMLongBench-Doc (Ma et al., 2024), and MP-DocVQA (Tito et al., 2022). Main results use the validation splits of DUDE and MP-DocVQA and the evaluation sets of MMDA and MMLongBench-Doc; official DUDE and MP-DocVQA test results are reported in Appendix B.1. We report ANLS (Average Normalized Levenshtein Similarity) for DUDE, ANLS and accuracy for MP-DocVQA, ACNLS and LLM-judged AIC accuracy for MMDA as in (Xu et al., 2026), and accuracy/F1 for MMLongBench-Doc.

Implementation details. We use Qwen3-VL-4B (Qwen Team, 2025) as the backbone. To strengthen the uncompressed DocVQA baseline, we fine-tune it on datasets with available training splits, including DUDE and MP-DocVQA; for datasets without training splits, we use the DUDE-fine-tuned model. HiToc introduces no additional training, and all inference experiments are conducted on AMD Instinct MI308X GPUs. Each document image produces 768 visual tokens ($H/32 \times W/32$), and we set $\tau = 0$, $L_p = 20$, $L_s = 5$, and $L_d = 20$. For the final active budget, we use $k_d = 80$ on DUDE and MMLongBench-Doc, 75 on MMDA, and 70 on MP-DocVQA.

Table 1: Performance comparison of different modules of HiToc on various DocQA benchmarks.

	DUDE			MMDA				MMLongBench-Doc				MP-DocVQA			
Methods	ANLS	Avg. Tokens	Retain Ratio	ACNLS	AIC Acc	Avg. Tokens	Retain Ratio	Acc	F1 Score	Avg. Tokens	Retain Ratio	ANLS	Acc	Avg. Tokens	Retain Ratio
Baseline	0.5902	4277.3	100%	0.5627	62.89%	5231.7	100%	27.76%	23.7%	36485.2	100%	0.8382	72.49%	4220.4	100%
+BBTM	0.5853	3342.3	78.1%	0.5642	63.14%	4267.4	81.6%	27.62%	22.9%	27999.6	76.7%	0.8306	71.16%	3387.0	80.3%
+BBTM+APTA	0.5831	918.0	21.5%	0.5620	63.02%	1020.0	19.5%	27.37%	22.5%	1644.2	4.5%	0.8286	70.91%	1099.8	26.1%
HiToc	**0.5791**	**734.4**	**17.2%**	**0.5525**	**61.98%**	**765.0**	**14.6%**	**26.63%**	**22.5%**	**1315.2**	**3.6%**	**0.8275**	**70.99%**	**769.8**	**18.2%**

4.2 Main Results

4.2.1 Comparisons with the uncompressed baseline.

Tab. 1 shows that HiToc substantially reduces the visual-token budget with limited quality loss across four benchmarks. On DUDE, MMDA, and MP-DocVQA, it retains only 14.6%–18.2% of the original tokens while preserving roughly 98% of baseline performance, with these settings keeping the final active visual budget close to one page (about 768 tokens). On MMLongBench-Doc, where documents are longer and evidence more dispersed, HiToc reduces the average token count from 36,485.2 to 1,315.2 (3.6% retention, about two pages) while preserving over 95% of baseline accuracy. These results show that HiToc remains effective from moderate-length to very long documents.

Tab. 1 also shows the role of each module. BBTM provides moderate early compression with little quality change, APTA contributes the largest reduction through question-conditioned page-to-token selection, and dynamic KV-cache compression further shrinks the active visual context during decoding with only limited additional loss.

4.2.2 Comparisons with state-of-the-art methods

We compare HiToc with three groups of baselines: *LLM-stage methods* (FastV (Chen et al., 2024), DivPrune (Alvar et al., 2025), and SparseVLM (Zhang et al., 2025b)), *ViT-stage methods* (VisionZip (Yang et al., 2025) and PruMerge (Shang et al., 2025)), and *document-specific methods* (TokenCorr (Zhang et al., 2024) and DocPrune (Choi et al., 2026)). Since these methods operate at different stages, we compare them under stage-aligned retention budgets. TokenCorr and DocPrune are reimplemented under the same setup as HiToc, more details can be referred to Appendix. B.3.

Comparisons with visual-only compression methods. Tab. 2 reports two stage-aligned comparisons for VisionZip and PruMerge. Under the *ViT-stage budget*, we compare them with BBTM at matched ratios, since all three methods operate before question-conditioned LLM processing. Under the *final-stage budget*, we run the same methods at ratios matched to full HiToc.

At the ViT stage, BBTM outperforms both methods on all four benchmarks. Under the final-stage budget, forcing VisionZip or PruMerge to match HiToc's retention ratio causes severe degradation. This suggests that for document QA, strong compression should not be concentrated entirely before question-conditioned processing.

Comparisons with general-purpose compression methods. At matched final retention ratios, HiToc outperforms FastV, DivPrune, and SparseVLM on all four benchmarks. This indicates that generic LLM-stage pruning alone is insufficient for long-document QA, where compression must also preserve page structure and complementary cross-page evidence.

Comparisons with document-specific compression methods. Tab. 2 further compares HiToc with TokenCorr and DocPrune. HiToc consistently achieves the best performance, which we attribute to its stage-matched design. TokenCorr reduces redundancy through token correlations but is not explicitly question-conditioned, while DocPrune performs substantial pruning before question-document interaction. In contrast, HiToc preserves layout cues after visual encoding, uses question-conditioned page-to-token selection to reduce sink-induced cross-page competition, and further adapts visual access during decoding.

4.3 Ablation Study

We further analyze the design choices of the three core components in HiToc.

Table 2: Comparison with state-of-the-art compression methods at ViT-stage and final-stage budgets.

Budget	Method	DUDE		MMDA			MMLongBench-Doc			MP-DocVQA		
		ANLS	Ret.	ACNLS	AIC-Acc	Ret.	Acc.	F1	Ret.	ANLS	Acc.	Ret.
ViT	PruMerge	0.4948	78.1%	0.3642	41.11%	81.6%	24.05%	17.96%	76.7%	0.7414	59.53%	80.3%
	VisionZip	0.5470	78.1%	0.4893	54.64%	81.6%	25.06%	19.96%	76.7%	0.7546	61.94%	80.3%
	BBTM	**0.5853**	78.1%	**0.5642**	**63.14%**	81.6%	**27.62%**	**22.90%**	76.7%	**0.8306**	**71.16%**	80.3%
Final	*ViT-stage methods*											
	PruMerge	0.2082	17.2%	0.0837	9.79%	14.6%	10.17%	3.33%	3.6%	0.1922	8.12%	18.2%
	VisionZip	0.2643	17.2%	0.1031	12.63%	14.6%	11.67%	5.05%	3.6%	0.2678	12.94%	18.2%
	LLM-stage methods											
	FastV	0.4554	17.2%	0.3816	39.30%	14.6%	17.74%	9.94%	3.6%	0.5690	41.74%	18.2%
	DivPrune	0.4641	17.2%	0.3346	36.86%	14.6%	20.33%	12.33%	3.6%	0.6309	43.18%	18.2%
	SparseVLM	0.5671	17.2%	0.5284	55.67%	14.6%	21.13%	14.86%	3.6%	0.7827	63.85%	18.2%
	Document-specific methods											
	TokenCorr	0.4366	20.81%	0.3788	37.63%	20.80%	22.00%	17.22%	3.56%	0.6135	42.07%	20.81%
	DocPrune	0.5081	21.39%	0.4434	48.97%	19.93%	22.90%	17.05%	3.52%	0.6317	50.90%	18.29%
	HiToc	**0.5791**	17.2%	**0.5525**	**61.98%**	14.6%	**26.63%**	**22.50%**	3.6%	**0.8275**	**70.99%**	18.2%

Table 3: Ablation study of the three HiToc components.

Modules	Strategies	DUDE	MMDA		MMLongBench-Doc		MP-DocVQA	
		ANLS	ACNLS	AIC-Acc	Acc	F1	ANLS	Acc
BBTM	DirectDrop	0.5680	0.5211	58.25%	27.44%	22.60%	0.7917	66.82%
	MergeToOne	0.5715	0.5251	58.12%	27.54%	22.81%	0.8051	68.32%
	Ours	**0.5853**	**0.5642**	**63.14%**	**27.62%**	**22.90%**	**0.8306**	**71.16%**
APTA	DirectChoose	0.5620	0.5230	56.44%	23.10%	16.65%	0.7889	65.66%
	Ours	**0.5831**	**0.5620**	**63.02%**	**27.37%**	**22.50%**	**0.8286**	**70.91%**
KV-Cache	SingleStage	0.5772	0.5477	60.95%	25.98%	21.84%	0.8155	69.65%
	ThreeStage	0.5787	0.5457	61.60%	26.14%	**22.56%**	0.8263	70.77%
	Ours	**0.5791**	**0.5525**	**61.98%**	**26.63%**	22.50%	**0.8275**	**70.99%**

Effectiveness of BBTM. The first block of Tab. 3 compares BBTM with two simpler alternatives: DirectDrop, which removes all background tokens, and MergeToOne, which replaces each background run with a single token. BBTM performs best on all four benchmarks, indicating that background regions should be compressed in a conservative and structure-aware manner rather than removed aggressively.

We further conduct a no-reinsertion ablation, in which the merged background tokens are moved behind all foreground tokens while keeping their positional encodings unchanged. On DUDE, this reduces ANLS to 0.5738. The drop suggests that background tokens still provide useful layout cues, and that preserving their relative placement with respect to foreground content is important for document understanding. Overall, these results validate the effectiveness of both parts in BBTM: background token merging and position-aware reinsertion.

Effectiveness of APTA. The second block of Tab. 3 compares APTA with *DirectChoose*, which directly performs global token selection from attention scores without page-level budgeting. For fairness, both methods retain the same number of tokens for each sample. APTA consistently outperforms *DirectChoose* on all datasets, showing the advantage of hierarchical selection over a single global ranking. By allocating page-specific budgets before within-page selection, APTA reduces cross-page competition and makes token selection less sensitive to attention sinks, helping preserve complementary evidence from less salient pages. Additional qualitative examples are provided in Appendix C.2.

To further study page-level budget allocation, we introduce an exponent coefficient λ to control retention ratios on lower-scoring pages, while keeping the retention ratio of the highest-scoring page at 1:

$$\rho_p = \left(\frac{s_p}{\max_{p'} s_{p'}}\right)^{\lambda}, \tag{8}$$

where $\lambda < 1$ yields a flatter allocation and $\lambda > 1$ a sharper one. When $\lambda = 1$, the formulation reduces to Eq. 6, the default setting in HiToc.

The results in Fig. 3 (left) show that reducing λ below 1 does not improve ANLS, likely because the additional tokens from lower-scoring pages introduce more irrelevant information. Increasing λ above 1 also hurts performance, as overly sharp page allocation raises the risk of discarding useful cross-page evidence. Overall, $\lambda = 1$ gives the best balance between noise suppression and evidence preservation.

We also study the effect of the prefilling layer L_p used by APTA. Since Qwen3-VL-4B has 36 decoder layers, we evaluate L_p from 5 to 30. As shown in Fig. 3 (right), the best result is obtained at $L_p = 20$, suggesting that an intermediate layer provides a better trade-off between redundancy reduction and information preservation than either shallow or deep layers.

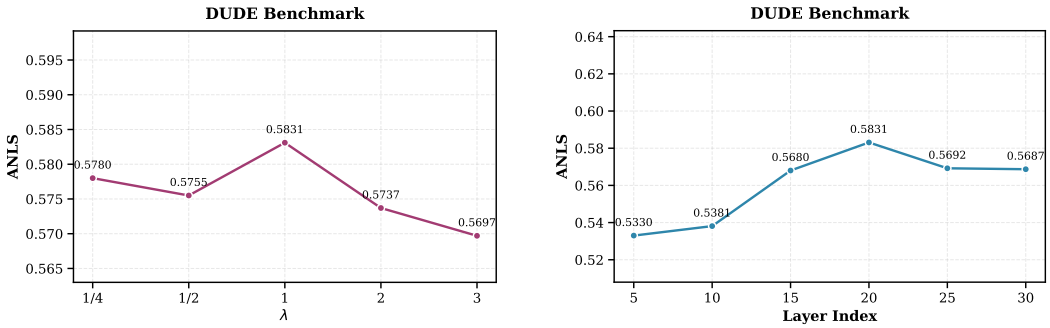

Figure 3: APTA ablations on DUDE. Left: page-allocation exponent λ. Right: prefilling layer L_p.

Effectiveness of dynamic KV-cache compression. Based on the layer analysis above, we set $L_s = 5$ and $L_d = 20$ for the shallow and deep stages of dynamic KV-cache compression. We compare this design with two alternatives: *SingleStage*, which performs KV compression only once at $L_s = 5$, and *ThreeStage*, which adds an intermediate pruning stage at $L_m = 15$ and uses $(L_s = 5, L_m = 15, L_d = 25)$, as reported in the third block of Tab. 3.

Multi-stage KV compression consistently outperforms the *SingleStage* variant across all settings. This supports our motivation that valuable visual context changes over both decoder depth and decoding step, making a single pruning step insufficient to capture dynamic relevance during generation.

Compared with *ThreeStage*, our final two-stage design achieves better overall performance, with the only exception being a slightly lower F1 score on MMLongBench-Doc (22.50% vs. 22.56%). More importantly, the two-stage variant requires fewer pruning operations and therefore introduces less latency and computational overhead, which achieves a better effectiveness–efficiency trade-off.

4.4 Efficiency Analysis

We evaluate inference efficiency with batch size 1 per GPU, using the same model checkpoint, BF16 precision, FlashAttention-2, and decoding configuration (max_new_tokens=256) for all methods. Qwen3-VL preprocessing is excluded because it is shared across all settings, and all results are averaged over three runs. To ensure fair comparison, we group baselines by the stage at which compression is applied. Specifically, we compare ViT-stage methods with BBTM at matched visual-token retention ratios in Tab. 4, and compare HiToc with LLM-stage methods on LLM inference latency in Tab. 5.

ViT-stage overhead. BBTM is designed to reduce the visual tokens passed to the LLM rather than accelerate the vision encoder itself. Its patch-labeling pipeline runs asynchronously with ViT inference, so only the lightweight token-merging step lies on the critical path. As shown in Tab. 4, BBTM adds only 1.40–2.37% visual-stage latency over the uncompressed baseline and remains faster than VisionZip and PruMerge at matched retention ratios. These results indicate that BBTM introduces only marginal front-end overhead while providing a more compact visual sequence.

Table 4: ViT-stage latency in minutes at matched visual-token retention ratios. Overhead is the relative latency increase of BBTM over the uncompressed baseline.

Dataset	Ret.	Baseline	VisionZip	PruMerge	**BBTM**	Overhead
DUDE	78.1%	142.69	158.09	160.61	**144.75**	1.44%
MMDA	81.6%	74.29	85.12	118.98	**75.33**	1.40%
MMLongBench-Doc	76.7%	457.34	501.53	712.99	**467.82**	2.29%
MP-DocVQA	80.3%	217.61	233.39	235.05	**222.76**	2.37%

LLM inference latency. Since FastV, DivPrune, and SparseVLM do not modify the visual stage, we compare them with HiToc only on LLM-stage latency in Tab. 5. Relative to the baseline, HiToc speeds up prefilling by $1.74\times$–$3.77\times$, decoding by $1.31\times$–$4.88\times$, and total LLM inference by $1.50\times$–$4.05\times$. The largest gain appears on MMLongBench-Doc, where latency drops from 424.46 to 104.76 minutes, indicating that the benefit of hierarchical compression grows with document length.

Table 5: LLM-stage inference latency in minutes over three runs.

Dataset	Stage	Baseline	FastV	DivPrune	SparseVLM	**HiToc**	**Speedup**
DUDE	Prefilling	56.24±0.51	28.41±0.12	42.07±0.26	41.35±0.25	26.97±1.81	2.09×
	Decoding	109.41±1.80	66.50±4.02	70.73±3.76	76.48±4.18	83.44±3.52	1.31×
	Total	**165.65**	**94.91**	**112.80**	**117.83**	**110.41**	**1.50×**
MMDA	Prefilling	9.94±0.09	4.81±1.10	6.92±1.03	7.35±1.41	4.64±0.01	2.14×
	Decoding	26.29±0.15	16.64±1.41	8.64±1.61	8.60±2.63	9.70±0.17	2.71×
	Total	**36.23**	**21.45**	**15.56**	**15.95**	**14.34**	**2.53×**
MMLongBench-Doc	Prefilling	295.39±7.22	125.96±5.85	143.65±7.47	240.66±6.19	78.33±5.37	3.77×
	Decoding	129.07±0.61	18.38±1.21	16.16±0.80	20.12±2.18	26.43±2.54	4.88×
	Total	**424.46**	**144.34**	**159.81**	**260.78**	**104.76**	**4.05×**
MP-DocVQA	Prefilling	45.78±0.04	27.92±1.51	30.30±1.64	40.22±1.96	26.25±1.82	1.74×
	Decoding	81.43±1.28	99.18±1.33	66.95±3.63	62.31±4.77	50.07±1.11	1.63×
	Total	**127.21**	**127.10**	**97.25**	**102.53**	**76.32**	**1.67×**

Memory and TFLOPs. We also compare average incremental GPU memory usage, excluding the approximately 8.4 GiB of model-resident memory, together with average TFLOPs. As shown in Tab. 6, HiToc reduces TFLOPs by 25.61–39.48% and incremental GPU memory by 2.42–16.37%. The largest reductions again occur on MMLongBench-Doc, where memory drops from 9.50 to 7.95 GiB and TFLOPs from 342.02 to 206.99. This trend is consistent with the latency results and further shows that HiToc becomes increasingly beneficial as document length grows.

Table 6: Incremental GPU memory and computational cost.

	Memory (GiB)			TFLOPs		
Dataset	Baseline	**HiToc**	Red.	Baseline	**HiToc**	Red.
DUDE	1.22	**1.12**	8.20%	42.79	**29.13**	31.93%
MMDA	1.49	**1.38**	7.61%	52.51	**35.49**	32.41%
MP-DocVQA	1.24	**1.21**	2.42%	42.22	**31.41**	25.61%
MMLongBench-Doc	9.50	**7.95**	16.37%	342.02	**206.99**	39.48%

5 Conclusion

HiToc addresses long-document QA with a training-free hierarchical token compression strategy across visual encoding, prefilling, and decoding. It preserves document structure, allocates question-relevant evidence more effectively, and adapts active visual context during generation through background token merging, page-to-token allocation, and dynamic visual KV selection. Across four benchmarks, HiToc reduces the final active visual-token budget to 3.6–18.2% of the original input while preserving 94.94–98.72% of baseline performance and improving inference efficiency. These results show that stage-aware visual compression is an effective strategy for long-document QA.

AI USE STATEMENT

We used generative AI tools to assist with language editing, manuscript organization, and checks for grammatical errors, numerical inconsistencies, and alignment between claims and experimental evidence. The authors reviewed the AI-generated suggestions and verified the revisions incorporated into the manuscript. The authors take full responsibility for the final content of this work, including its methods, results, and conclusions.

REFERENCES

Ahmed N. Albatineh and Magdalena Niewiadomska-Bugaj. Correcting jaccard and other similarity indices for chance agreement in cluster analysis. *Advances in Data Analysis and Classification*, 5 (3):179–200, 2011. doi: 10.1007/s11634-011-0090-y.

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401, 2025. URL https://api.semanticscholar.org/CorpusID:276775957.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JroZRaRw7Eu.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=ayi7qezU87.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models, 2024.

Yew Ken Chia, Liying Cheng, Hou Pong Chan, Chaoqun Liu, Maojia Song, Sharifah Mahani Aljunied, Soujanya Poria, and Lidong Bing. M-longdoc: A benchmark for multimodal super-long document understanding and a retrieval-aware tuning framework. *ArXiv*, abs/2411.06176, 2024. URL https://api.semanticscholar.org/CorpusID:273963255.

Joonmyung Choi, Sanghyeok Lee, Jongha Kim, Sehyung Kim, Dohwan Ko, Jihyung Kil, and Hyunwoo J Kim. Docprune: Efficient document question answering via background, question, and comprehension-aware token pruning. *arXiv preprint arXiv:2604.22281*, 2026.

Zizhong Ding, Junxian Li, Kai Liu, Shaoqiu Zhang, Xiao Xiao, Linghe Kong, and Yulun Zhang. ET-Prune: Evidence-aware dynamic budgeting for visual token pruning in text-rich MLLMs, 2026. URL https://arxiv.org/abs/2608.01979.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models. In *International Conference on Learning Representations*, volume 2025, pp. 61424–61449, 2025.

Hao Feng, Qi Liu, Hao Liu, Jingqun Tang, Wengang Zhou, Houqiang Li, and Can Huang. Docpedia: Unleashing the power of large multimodal model in the frequency domain for versatile document understanding. *Science China Information Sciences*, 67(12):220106, 2024.

Yefei He, Feng Chen, Jing Liu, Wenqi Shao, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. Zipvl: Efficient large vision-language models with dynamic token sparsification and kv cache compression. *ArXiv*, abs/2410.08584, 2024. URL https://api.semanticscholar.org/CorpusID:273323458.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr F. Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *ArXiv*, abs/2404.14469, 2024. URL https://api.semanticscholar.org/CorpusID:269303164.

Youwei Liang, Chongjian GE, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. EVit: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=BjyvwnXXVn_.

Wenhui Liao, Jiapeng Wang, Hongliang Li, Chengyu Wang, Jun Huang, and Lianwen Jin. Doclayllm: An efficient multi-modal extension of large language models for text-rich document understanding. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4038–4049, 2025. URL https://api.semanticscholar.org/CorpusID:277112782.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, Pan Zhang, Liangming Pan, Yu-Gang Jiang, Jiaqi Wang, Yixin Cao, and Aixin Sun. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations, 2024. URL https://arxiv.org/abs/2407.01523.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2200–2209, 2021.

Mor Shpigel Nacson, Aviad Aberdam, Roy Ganz, Elad Ben Avraham, Alona Golts, Yair Kittenplon, Shai Mazor, and Ron Litman. Docvlm: Make your vlm an efficient reader. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29005–29015, 2025. URL https://api.semanticscholar.org/CorpusID:274656409.

Qwen Team. Qwen3-VL technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=kR95DuwwXHZ.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *ICCV*, 2025.

Irwin Sobel and G. M. Feldman. An isotropic 3×3 image gradient operator, 1990. URL https://api.semanticscholar.org/CorpusID:59909525.

Rubèn Pérez Tito, Dimosthenis Karatzas, and Ernest Valveny. Hierarchical multimodal transformers for multi-page docvqa. *ArXiv*, abs/2212.05935, 2022. URL https://api.semanticscholar.org/CorpusID:254563929.

Jordy Van Landeghem, Rubèn Tito, Łukasz Borchmann, Michał Pietruszka, Dawid Jurkiewicz, Rafał Powalski, Paweł Józiak, Sanket Biswas, Mickaël Coustaty, and Tomasz Stanisławek. Icdar 2023 competition on document understanding of everything (dude). In *Proceedings of the ICDAR 2023*, 2023.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024.

Bangrui Xu, Qihang Yao, Zirui Tang, Xuanhe Zhou, Yeye He, Shihan Yu, Qianqian Xu, Bin Wang, Guoliang Li, Conghui He, and Fan Wu. Modora: Tree-based semi-structured document analysis system. *ACM SIGMOD*, 2026.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802, 2025. URL https://api.semanticscholar.org/CorpusID:274514545.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, et al. mplug-docowl: Modularized multimodal large language model for document understanding. *arXiv preprint arXiv:2307.02499*, 2023a.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, et al. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 2841–2858, 2023b.

Hui Zeng, Daming Zhao, Pengfei Yang, Wenxuan Hou, Tianyang Zheng, Hui Li, Weiye Ji, and Jidong Zhai. Lethe: Layer- and time-adaptive KV cache pruning for reasoning-intensive LLM serving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. doi: 10.1609/aaai.v40i33.40036. URL https://ojs.aaai.org/index.php/AAAI/article/view/40036.

Hao Zhang, Mengsi Lyu, Bo Huang, Yulong Ao, and Yonghua Lin. TrimTokenator-LC: Towards adaptive visual token pruning for large multimodal models with long contexts, 2025a. URL https://arxiv.org/abs/2512.22748.

Renshan Zhang, Yibo Lyu, Rui Shao, Gongwei Chen, Weili Guan, and Liqiang Nie. Token-level correlation-guided compression for efficient multimodal document understanding. *arXiv preprint arXiv:2407.14439*, 2024.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=80faIPZ67S.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2o: heavy-hitter oracle for efficient generative inference of large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023.

A Benchmarks and Evaluation Scope

A.1 DUDE

DUDE (Document Understanding Dataset and Evaluation) (Van Landeghem et al., 2023) is a large-scale multi-page DocQA benchmark containing over 5,000 real-world documents from multiple industries and domains, together with approximately 41,500 question-answer pairs. The dataset covers diverse document formats, including both born-digital files and scanned materials with complex layouts, and the documents typically consist of six pages. The dataset is divided into training, validation, and test splits with 3,010/749/1,215 documents, corresponding to 23,728/6,315/11,448 QA pairs, respectively. The ground-truth annotations for the test split are not publicly released, and test-set performance can only be obtained through submission to the official evaluation server. Moreover, DUDE is intentionally designed with distribution shifts between the training and test sets to evaluate model generalization. Following the official evaluation protocol, we adopt ANLS (Average Normalized Levenshtein Similarity) as the evaluation metric.

A.2 MMDA

MMDA (Xu et al., 2026) is a newly released DocQA benchmark comprising 1,065 QA pairs over 537 documents for evaluation. Specifically, the dataset contains 444 text-based questions, 250 hybrid questions involving both text and non-text elements, 159 hierarchy-related questions, 151 location-based questions, and 61 questions targeting formatted text information. MMDA covers a broad range of complex single-column and multi-column layouts with rich multimodal content, requiring stronger reasoning over document structure, hierarchy, and cross-modal evidence. Following the official evaluation protocol, we report both ANLS and AIC-Acc. In particular, the ANLS metric used in MMDA is the ACNLS variant proposed in Xu et al. (2026), which assigns full credit when the predicted answer contains the reference answer. The AIC-Acc metric is based on LLM judgment to account for semantic equivalence across varied expressions, assigning 1 to correct answers and 0 otherwise.

A.3 MMLongBench-Doc

MMLongBench-Doc (Ma et al., 2024) is a highly challenging benchmark proposed in 2024 for long-context document understanding. It comprises 1,082 expert-annotated questions constructed upon 135 lengthy PDF-formatted documents across 7 diverse domains. Each document contains 47.5 pages and 21,214 textual tokens on average. The benchmark includes 494 single-page questions, 365 cross-page questions, and 223 unanswerable questions, making it particularly challenging for evaluating long-range reasoning and evidence aggregation capabilities in vision-language models. Following Ma et al. (2024), we report accuracy and F1-Score as the evaluation metrics.

A.4 MP-DocVQA

MP-DocVQA (Tito et al., 2022) is a multi-page extension of the widely-used DocVQA (Mathew et al., 2021) dataset, designed to evaluate visual question answering over documents spanning multiple pages. The dataset contains approximately 46,000 questions over $\sim$6,000 documents from various industries, with each document capped at a maximum of 20 pages to reflect common real-world scenarios. Similar to DUDE, the dataset is divided into training, validation, and test splits, with the ground-truth annotations of the test split not publicly available. Following the official evaluation protocol, we report both Accuracy and ANLS in our experiments.

B Additional Experimental Results

B.1 Official Test-Split Results

To further validate the robustness and generalization capability of HiToc, we conducted additional experiments on the official test splits of the DUDE and MP-DocVQA benchmarks. The results are summarized in Tab. 7.

Table 7: Results on the official DUDE and MP-DocVQA test splits.

Dataset	Stage	ANLS	Mean tokens	Retention
DUDE	Uncompressed	0.5763	4,222.5	100%
	BBTM	0.5690	3,250.1	76.97%
	BBTM+APTA	0.5680	974.2	23.07%
	Full HiToc	0.5604	779.4	18.46%
MP-DocVQA	Uncompressed	0.8465	3,874.7	100%
	BBTM	0.8370	3,127.4	80.71%
	BBTM+APTA	0.8345	1,079.6	27.86%
	Full HiToc	0.8305	755.72	19.50%

As is shown, HiToc maintains its exceptional efficiency-accuracy trade-off on unseen data, exhibiting performance trends highly consistent with those observed on the validation splits. Specifically, on the DUDE test set, HiToc retains 97.24% of the baseline ANLS while compressing the visual sequence to merely 18.46% of the original token count. Similarly, on the MP-DocVQA test set, the method preserves 98.11% of the baseline ANLS with only 19.50% of the tokens retained. This consistency across different data distributions strongly corroborates that our hierarchical compression strategy is not overfitted to specific validation samples but possesses strong generalization capabilities, making it a reliable solution for real-world document understanding tasks where the input diversity is high.

B.2 Results on Qwen3-VL-8B

Table 8: Transfer to Qwen3-VL-8B. This evaluates a larger backbone within the Qwen3-VL family. Accuracy and F1 are percentages.

	DUDE	MMDA		MMLongBench		MP-DocVQA	
Stage	ANLS	ACNLS	AIC-Acc.	Acc.	F1	ANLS	Acc.
Uncompressed	0.5987	0.5950	70.10	29.73	25.58	0.8655	77.21
BBTM	0.5934	0.5958	67.53	30.33	25.96	0.8559	75.61
BBTM+APTA	0.5862	0.5928	67.14	29.12	24.93	0.8457	74.30
Full HiToc	0.5877	0.5851	65.98	28.83	25.11	0.8509	74.86

As is shown in Tab. 8, HiToc maintains its exceptional efficiency-accuracy trade-off on unseen data, exhibiting performance trends highly consistent with those observed on the validation splits. Specifically, on the DUDE test set, HiToc retains 97.24% of the baseline ANLS while compressing the visual sequence to merely 18.46% of the original token count. Similarly, on the MP-DocVQA test set, the method preserves 98.11% of the baseline accuracy with only 19.50% of the tokens retained. This consistency across different data distributions strongly corroborates that our hierarchical compression strategy is not overfitted to specific validation samples but possesses strong generalization capabilities, making it a reliable solution for real-world document understanding tasks where input diversity is high.

B.3 More Comparisons on Document-specific Methods

B.3.1 Implementation Details of Document-Specific Baselines

Official implementations of TokenCorr (Zhang et al., 2024) and DocPrune (Choi et al., 2026) were not publicly available at the time of our experiments. We therefore implemented both methods based on their papers and adapted them to the Qwen3-VL-4B backbone used by HiToc. Below, we summarize the backbone-specific adaptations, compression settings, and the resulting token budgets

For TokenCorr, our implementation follows the original pipeline of correlation-based redundancy estimation, token sampling, and neighborhood aggregation. We set the cosine-similarity threshold to 0.85, the redundancy-count threshold to 8, and the aggregation neighborhood size to 4. In our adaptation to Qwen3-VL, compression is applied at decoder layer 20 for alignment with HiToc. Since Qwen3-VL does not provide a CLS feature, we estimate token correlations directly from token features and use their mean feature as a proxy for the original CLS-based importance signal. Each

Table 9: Stage-wise latency comparison with document-specific baselines on four benchmarks (minutes).

Dataset	Method	Retrieval	ViT	Prefill	Decode	**LLM**	**Total**
DUDE	TokenCorr	–	46.33	74.30	75.42	149.72	196.04
	DocPrune	285.82	59.62	10.56	72.17	82.74	428.17
	HiToc	–	**144.75**	**26.97**	**83.44**	**110.41**	**255.16**
MMDA	TokenCorr	–	45.67	11.34	8.33	19.67	65.34
	DocPrune	66.21	26.48	1.44	10.35	11.79	104.49
	HiToc	–	**75.33**	**4.64**	**9.70**	**14.34**	**89.67**
MP-DocVQA	TokenCorr	–	36.77	60.64	57.82	118.45	155.22
	DocPrune	248.66	85.85	7.69	57.26	64.95	399.46
	HiToc	–	**222.76**	**26.25**	**50.07**	**76.32**	**299.08**
MMLongBench-Doc	TokenCorr	–	410.90	240.89	14.34	255.23	666.13
	DocPrune	414.30	19.67	2.38	15.02	17.40	451.37
	HiToc	–	**467.82**	**78.33**	**26.43**	**104.76**	**572.58**

page is represented by a global overview and up to four local crops, each at a resolution of 448×448. When four local crops are used, their combined pixel budget (802,816 pixels) is comparable to HiToc's per-page input budget (786,432 pixels), with an additional global view preserving full-page context.

For DocPrune, we follow the main configuration in the original paper. Specifically, we use ColPali (Faysse et al., 2025) to retrieve the top four pages and setting the QA background threshold to $\tau_{\text{bg}} = 0.8$, the comprehension threshold to $\tau_{\text{comp}} = 45$, and the attention threshold to $\tau_{\text{att}} = 0.075$. The question-relevance threshold τ_{qst} is tuned separately on each benchmark to control the final token retention ratio, and is set to 0.1, -1.6, 1.65, and 2.8 for DUDE, MMDA, MP-DocVQA, and MMLongBench-Doc, respectively.

Since both methods involve multiple selection and compression steps, their final token budgets cannot be matched exactly to a prescribed target. We therefore calibrate their parameters to achieve approximately 20% retention on DUDE, MMDA, and MP-DocVQA, and approximately 3.6% on MMLongBench-Doc.

B.3.2 Latency Comparison with Document-Specific Baselines

Tab. 9 reports stage-wise latency for TokenCorr, DocPrune, and HiToc. The total includes external retrieval, when applicable, together with visual encoding, prefilling, and decoding.

Comparison with TokenCorr. TokenCorr and HiToc show different efficiency profiles across the pipeline. Despite comparable overall pixel budgets, TokenCorr achieves lower visual-encoding latency on all four benchmarks. Its fixed-resolution crops shorten the per-image attention sequence length and produce more regular input shapes, reducing both attention cost and shape-dependent compilation overhead.

In contrast, under broadly matched LLM input budgets and the same decoding setup, HiToc is consistently both faster and more accurate than TokenCorr in the LLM stage. This is especially clear on MMLongBench-Doc, where HiToc reduces LLM-stage latency from 255.23 to 104.76 minutes, while improving accuracy from 22.00% to 26.63% and F1 from 17.22% to 22.50%. In other words, although TokenCorr is more efficient at visual encoding because of regular and smaller patch images, HiToc shows a clear advantage once computation shifts to the language model.

Comparison with DocPrune. DocPrune reduces downstream computation through top-4 page retrieval and early token pruning, but introduces substantial retrieval overhead. Including retrieval, HiToc achieves lower total latency on DUDE and MP-DocVQA while also maintaining higher answer quality. On MMDA, HiToc remains competitive in total latency (89.67 vs. 104.49 minutes) and still performs substantially better in answer quality.

The trade-off is most evident on MMLongBench-Doc. Restricting each long document to four retrieved pages sharply reduces DocPrune's visual and LLM computation, but retrieval alone takes 414.30 minutes, accounting for 91.8% of the total latency. As a result, DocPrune remains faster overall than HiToc on this benchmark, but with lower accuracy and F1 by 3.73 and 5.45 points, respectively. This is consistent with the risk that aggressive page filtering may discard evidence distributed across long documents. By avoiding an external retrieval stage and progressively refining visual context across pages, HiToc achieves a stronger balance between efficiency and answer quality.

C Hyperparameter Ablations and Detailed Analyses

C.1 Bottom-up Background Token Merging

Connected-component filtering. We provide additional implementation details for the text-aware patch labeling process. After binarizing the Sobel response, we apply morphological dilation with a 3×3 kernel to connect nearby edge fragments. Connected components are then extracted using OpenCV's default 8-connectivity setting. Components smaller than 150 pixels are discarded to suppress isolated noisy responses before constructing candidate text regions. Bounding boxes are generated from the remaining components and converted into the binary masks used to form the saliency map S. Patch labels are subsequently computed from S following Eq. 3.

Sensitivity to the foreground-ratio threshold. Tab. 10 analyzes the effect of the BBTM threshold τ on DUDE. This threshold determines whether a patch is labeled as foreground according to its text-space ratio. Larger τ values make the criterion stricter, so patches with only limited text coverage are more likely to be treated as background and merged. The best ANLS, 0.5853, is achieved at $\tau = 0$. Using positive thresholds lowers ANLS to 0.5792–0.5845, suggesting that a patch should be preserved whenever it contains any text-related content. In this way, $\tau = 0$ is a simple and effective default setting in addition to yielding the best performance.

Table 10: Sensitivity of BBTM ANLS to the threshold τ on DUDE.

$\tau = 0$	0.01	0.03	0.05	0.10
0.5853	0.5845	0.5805	0.5816	0.5792

C.2 APTA: Page Allocation and Cross-Page Evidence

C.2.1 Comparison with Fixed Top-k Page Selection

Tab. 11 compares APTA with fixed top-k page selection under the same setting, with BBTM enabled and dynamic KV compression disabled. As expected, increasing k from 1 to 3 generally improves performance, but this comes with a substantial increase in retained tokens. Moreover, the performance gains are often marginal and inconsistent across benchmarks.

In contrast, APTA achieves a clearly better token–quality trade-off. Relative to the best-performing fixed top-k variants, APTA uses fewer tokens on MMDA (1020.0 vs. 1340.4), MMLongBench-Doc (1644.2 vs. 1782.6), and MP-DocVQA (1099.8 vs. 1487.7), while delivering the best performance on nearly all reported metrics. Even on DUDE, where top-3 attains a slightly higher ANLS than APTA (0.5863 vs. 0.5831), this advantage comes at a much larger token budget (1360.4 vs. 918.0), suggesting that the gain is obtained by retaining substantially more visual context rather than by using the budget more effectively. Overall, these results suggest that fixed top-k page retention is a coarse way to control visual context, whereas APTA utilizes the token budget more effectively by distributing it across pages and selecting tokens within each page. This allows lower-ranked pages to preserve supporting cross-page evidence without retaining entire pages. Moreover, APTA also provides a better starting point for the subsequent dynamic KV compression stage, which further reduces the final active visual budget to around one page while preserving strong answer quality, especially when compared with the top-$k = 1$ page selection setting.

Table 11: Comparison of top-k page selection methods with APTA, with BBTM enabled.

Model	DUDE		MMDA			MMLongBench-Doc			MP-DocVQA		
	ANLS	Avg. Tokens	ACNLS	AIC-Acc	Avg. Tokens	Acc	F1	Avg. Tokens	ANLS	Acc	Avg. Tokens
$k = 1$	0.4994	650.4	0.4593	50.45%	654.8	20.28%	10.83%	713.2	0.6670	55.99%	632.0
$k = 2$	0.5830	1069.5	0.5584	60.82%	1044.5	27.25%	22.06%	1320.2	0.8190	70.14%	1043.2
$k = 3$	**0.5863**	1360.4	0.5557	60.95%	1340.4	27.22%	22.32%	1782.6	0.8273	**71.02%**	1487.7
APTA	0.5831	918.0	**0.5620**	**63.02%**	1020.0	**27.37%**	**22.50%**	1644.2	**0.8286**	70.91%	1099.8

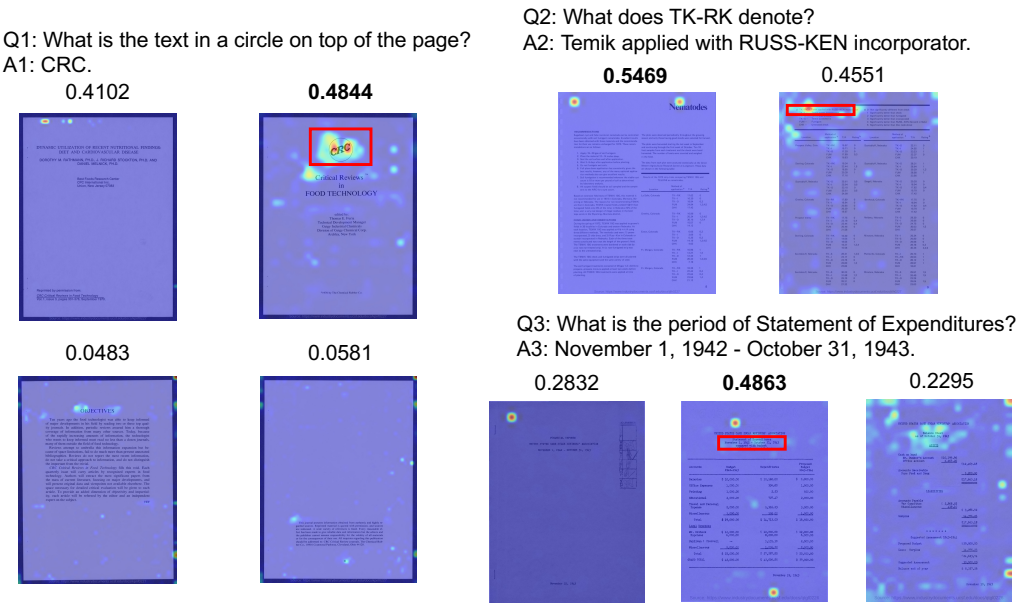

Figure 4: Visualization of page-level attention in APTA.

C.2.2 Qualitative Analysis

We provide several visualizations of the proposed APTA module on page-level relevance estimation. As is shown in Fig. 4, the APTA can accurately identify the target page containing the final answer and assigns relatively high attention weights to the target regions in most cases. Importantly, these visualizations further reveal several noteworthy observations.

First, attention sinks do arise, where information-sparse background tokens absorb disproportionately large attention values. However, APTA aggregates attention values at the page level, which smooths local noise and isolated peaks and leads to the correct page selection in Q1. In this way, by retaining all tokens within the target page, we preserve the contextual integrity essential for accurate reasoning. Simultaneously, keeping the attention sinks from other pages helps stabilize the global attention distribution, thereby ensuring a more robust and consistent inference process.

Second, in challenging cases like Q2, severe attention sinks on non-target pages (e.g., page #1) may dominate page selection, causing APTA to miss the true target page (Page #2) as the most relevant one. Despite this, since APTA is applied at intermediate layers, the semantically meaningful activations on the target page have already emerged during the forward pass. As a result, APTA can still assign sufficient attention to the truly answer-related regions on Page #2, allowing the key evidence to be retained and ultimately supporting correct answer prediction. In other words, even when APTA makes an imperfect page-level decision, it can still preserve the core evidence needed for answering. This robustness is a key advantage of the proposed design.

Third, Q3 demonstrates why page-wise preservation is preferable to global token ranking. In this example, the regions supporting the ground-truth answer are not the most prominent ones in the global attention distribution. As a result, a global top-k strategy would likely remove them in favor of more salient but less useful tokens. By contrast, APTA conducts the actual pruning operation within

each page, ensuring that tokens which are locally important to a specific page are preserved even if they are not globally top-ranked. This allows APTA to retain critical supporting evidence that would otherwise be lost under purely global selection.

C.2.3 Cross-Page Answering and Attention Distribution

We further assess whether HiToc preserves cross-page evidence on MMLongBench-Doc from two complementary perspectives: answering performance and page-level attention distribution. The goal is to verify that the PageToToken strategy in APTA does not overly bias token selection toward a small number of dominant pages, which could otherwise hurt questions requiring evidence aggregation across pages.

Cross-page answering performance. We evaluate single-page and cross-page questions separately. As shown in Tab. 12, HiToc retains 96.29% of the baseline accuracy and 95.36% of the baseline F1 on cross-page questions, compared with 94.13% and 94.11% on single-page questions. The slightly higher relative retention on the cross-page subset suggests that HiToc preserves cross-page answering ability well under compression, rather than disproportionately degrading questions that rely on multi-page evidence.

Page-attention distribution. We next analyze page-attention distribution to examine whether HiToc preserves the cross-page evidence access pattern of the uncompressed model. For normalized page-attention scores π_p, we use the effective number of attended pages,

$$N_{\text{eff}} = \frac{1}{\sum_p \pi_p^2}, \tag{9}$$

which can be interpreted as the effective number of pages receiving substantial attention: larger values indicate a broader and less concentrated page-attention distribution. We also report the number of pages receiving at least 5% of the total attention mass, together with the normalized attention variance. As shown in Tab. 13, cross-page questions exhibit broader page-attention distributions than single-page questions for both the uncompressed model and HiToc, as reflected by larger N_{eff}, more pages with non-trivial attention mass, and lower attention variance. Importantly, HiToc remains very close to the uncompressed baseline on all three statistics: for example, N_{eff} changes only from 5.88 to 5.99 for single-page questions and from 6.20 to 6.28 for cross-page questions. These results suggest that compression does not collapse attention onto a small set of dominant pages, but largely preserves the cross-page allocation pattern needed for multi-page evidence aggregation.

Taken together, Tab. 12 and 13 provide consistent evidence that HiToc preserves cross-page evidence from both outcome and behavior perspectives: it maintains most of the answering performance on cross-page questions, while preserving a page-attention spread similar to that of the uncompressed model. We attribute this capability primarily to APTA's page-to-token design, which allocates token budgets across pages before within-page selection, thereby reducing inter-page competition and helping retain complementary evidence from multiple pages.

Table 12: Single-page and cross-page answering on MMLongBench-Doc.

Question type	Metric	Uncompressed	HiToc	Quality retained
Single-page	Acc.	32.70%	30.78%	94.13%
	F1	33.46%	31.49%	94.11%
Cross-page	Acc.	14.82%	14.27%	96.29%
	F1	15.31%	14.60%	95.36%

Table 13: Page-attention statistics on single-page and cross-page questions in MMLongBench-Doc.

Question type	Model	Pages with $\geq 5\%$ mass	Norm. variance	N_{eff}
Single-page	Uncompressed	2.64	0.1632	5.88
	HiToc	2.66	0.1613	5.99
Cross-page	Uncompressed	2.94	0.1457	6.20
	HiToc	2.96	0.1446	6.28

C.3 Dynamic KV Compression: Budget Sensitivity and Cross-Depth Complementarity

We further analyze dynamic KV compression from two aspects: how sensitive performance is to the active visual KV budget, and whether visual KV selection differs across decoder depths.

Effect of active visual KV budget. We vary k_d from 50 to 90 while fixing the two selection depths at $(L_s, L_d) = (5, 20)$. Here, k_d denotes the fraction of post-APTA visual KV candidates activated at each selection stage.

Tab. 14 shows that performance is generally stable when k_d is between 70 and 90. When the active budget is reduced below 70%, the decline becomes more noticeable on MMDA ACNLS and MP-DocVQA ANLS, whereas DUDE ANLS remains comparatively stable across the sweep. The best value is somewhat dataset-dependent: DUDE and MMLongBench-Doc peak at around 80%, while MMDA peaks at 75 or 70, and MP-DocVQA metrics peak at 70%. Nevertheless, $k_d = 70$ provides a strong and well-balanced setting across all benchmarks, achieving near-best performance without benchmark-specific tuning. This suggests that a single default active budget can already work reliably in practice, reducing the need for extra parameter search especially on a new benchmark.

Complementarity across decoder depths. We next examine whether shallow and deep decoder stages activate similar visual KV entries. To measure their overlap, we employ chance-corrected Jaccard similarity (Albatineh & Niewiadomska-Bugaj, 2011) rather than raw set overlap, because both stages retain a relatively large fraction of candidates and would therefore exhibit non-trivial overlap even under random selection. By discounting this expected overlap, the metric provides a more reliable measure of whether the two depths truly select similar visual evidence. The resulting scores are 0.153 on DUDE, 0.175 on MMDA, 0.196 on MP-DocVQA, and 0.161 on MMLongBench-Doc. These modest corrected similarities indicate that shallow and deep decoder stages activate substantially different visual subsets beyond chance overlap, supporting the design intuition that different decoder depths benefit from complementary visual evidence rather than sharing a single fixed active set.

Table 14: Sensitivity to active KV retention k_d (%). Accuracy and F1 are percentages. Bold denotes the best value in each metric column.

	DUDE	MMDA		MMLongBench		MP-DocVQA	
k_d	ANLS	ACNLS	AIC-Acc.	Acc.	F1	ANLS	Acc.
90	0.5776	0.5482	**61.98**	26.18	22.44	0.8255	70.75
85	0.5768	0.5516	**61.98**	26.27	22.48	0.8260	70.72
80	**0.5791**	0.5519	61.86	**26.63**	22.50	0.8261	70.77
75	0.5767	0.5525	**61.98**	26.32	22.49	0.8267	70.91
70	0.5785	**0.5546**	61.60	26.61	**22.61**	**0.8275**	**70.99**
65	0.5764	0.5507	61.56	26.31	22.33	0.8250	70.68
60	0.5777	0.5496	61.47	26.25	22.42	0.8248	70.58
55	0.5780	0.5451	61.38	26.42	22.41	0.8236	70.48
50	0.5771	0.5457	61.22	26.28	22.37	0.8226	70.52

C.4 Answerability Analysis

Unanswerable questions are an important part of document question answering, since real-world documents do not always contain sufficient evidence for every query. Therefore, a reliable token compression method should preserve not only answer extraction ability on answerable questions, but also the model's ability to correctly recognize when no answer is supported by the document. To

assess this aspect, we evaluate HiToc on the two benchmarks in our suite that include unanswerable questions, and separately report answer extraction and unanswerability recognition in Tab. 15.

On DUDE, compression reduces answerable-question ANLS from 0.5787 to 0.5634 and unanswerable-question accuracy from 69.07% to 67.63%, retaining 97.36% and 97.92% of the respective baseline scores. On MMLongBench-Doc, unanswerable-question accuracy changes only slightly, from 38.57% to 38.12%. These small reductions across both answerable and unanswerable settings suggest that the compression configuration preserves most of the baseline's ability not only to extract answers when evidence is present, but also to abstain appropriately when supporting evidence is missing.

Table 15: Answerability analysis on DUDE and MMLongBench-Doc.

Dataset	Subset	Metric	Base	HiToc	Retained
DUDE	Answerable	ANLS	0.5787	0.5634	97.36%
DUDE	Unanswerable	Acc.	69.07%	67.63%	97.92%
MMLongBench-Doc	Unanswerable	Acc.	38.57%	38.12%	98.83%