DocSLM: A Small Vision-Language Model for Long Multimodal Document Understanding

Supplementary Material

Our supplementary materials contain Section S1: Additional Implementation Details, Section S2: Edge Deployment, and Section S3: Efficiency Analysis, Section S4: Additional Ablation Studies, Section S5: Additional Qualitative Results.

S1. Additional Implementation Details

S1.1. OCR Integration

Our OCR pipeline is built around a bounding-box alignment mechanism (Fig. F1) that enables consistent OCR integration under multi-crop processing [14] to handle documents of any size and shape. As illustrated in Fig. F2, each input page is first resized, padded, and subdivided into a grid of non-overlapping crops. OCR tokens detected on the original image must therefore be remapped to these crops in a geometrically consistent manner. Fig. F1 and F2 show 4 regions for simplicity; however, based on aspect ratio and resolution, the number of crops can range from 4-16 in our default setup. As visualized in Fig. F1, each OCR token is represented by a bounding box in original-image coordinates, normalized by the image width and height. A token is assigned to every crop whose bounding box overlaps with it. This supports one-to-many assignments when a word spans crop boundaries and handles empty crops. The resulting crop-aligned OCR lists are then fused with the hierarchical multimodal compression module. This alignment mechanism ensures that multimodal training receives consistent and spatially grounded OCR information, even under highly variable document layouts and multi-resolution patch configurations.

S1.2. Training Details

Table T1 summarizes the full five-stage training pipeline used to build our 2B-parameter model. The training strategy gradually transitions from large-scale noisy pretraining to highly curated downstream finetuning, while progressively increasing task difficulty and reducing learning rates. Pretrain 1 initializes the multimodal alignment by training the MLP adapter and hierarchical compressor on 1M weakly supervised image–text pairs using cross-attention–based fusion of SigLIP2[16] visual features and PaddleOCR[13] tokens. Pretrain 2 scales the same objective to a larger 3M corpus and unlocks the vision tower and multimodal compressor for joint optimization, improving cross-modal grounding. Pretrain 3 adapts the model to high-quality single-page document datasets (2M samples), introduces

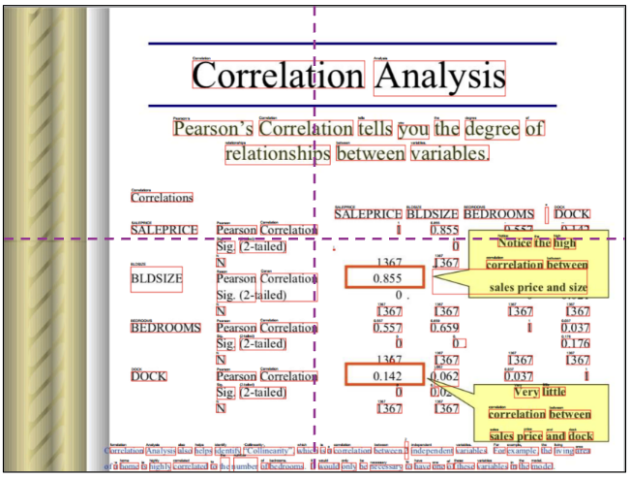


Figure F1. **OCR-to-crop assignment.** The OCR bounding boxes (red) are tested for overlap with the crop regions. An OCR token is assigned to a crop if its bounding boxes intersect, ensuring spatially consistent OCR alignment across crops.

Stage	Training Steps	Batch	Data Size	LR
Pretrain 1	3.0K	1.0K	1.00M	1×10^{-4}
Pretrain 2	9.0K	1.0K	3.00M	1×10^{-4}
Pretrain 3	2.4K	1.0K	2.00M	2×10^{-5}
Finetune 1	3.0K	256	0.58M	2×10^{-5}
Finetune 2	4.4K	256	0.18M	2×10^{-6}

Table T1. Each stage progressively adapts to more complex tasks, while the availability of high-quality data decreases.

early-layer OCR and visual compression, and begins tuning the language model to better handle structured document semantics. Finetune 1 transitions to the DocDownstream-1.0[10] mixture (0.58M examples) and trains under long-context settings, enabling robust reasoning over long documents while maintaining a manageable batch size via ZeRO-2 and gradient accumulation[9]. Finally, Finetune 2 introduces negative-pair supervision and multi-image document sequences, training the model to abstain on unsupported evidence and improving calibration in streaming settings.

Across all stages, we use bf16 precision and flash-attention [6]. This staged progression allows the model to retain broad generalization from large-scale pretraining while acquiring strong long-document reasoning capabilities from high-quality downstream data. To further boost computational and memory efficiency, we incorporate Liger

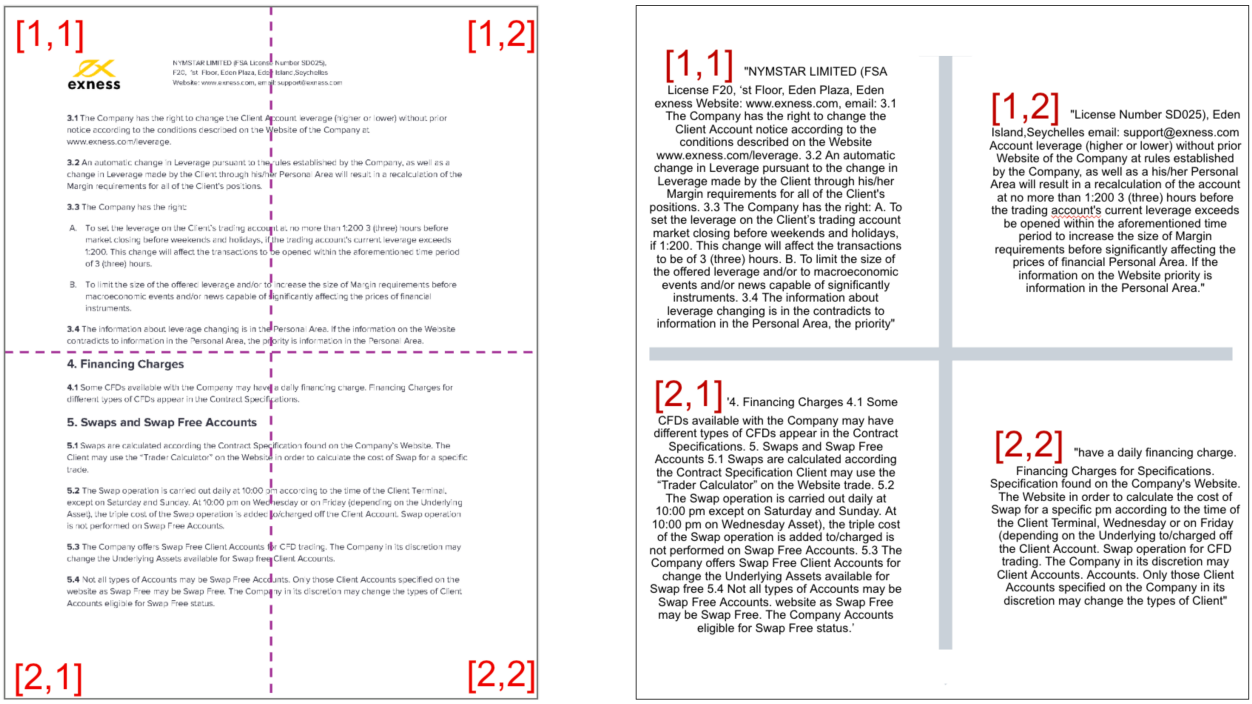


Figure F2. **Multi-crop OCR decomposition. (left)** Each page is first resized and padded, then dynamically divided into an aspect-ratio–dependent grid of overlapping crops. **(right)** OCR tokens are spatially redistributed to their corresponding crops, enabling localized grounding and improving fine-grained multimodal alignment.

Kernel [4], a lightweight optimization toolkit designed for large-scale model training. Liger provides high-performance fused operators and memory-aware execution strategies, such as combining sequential kernels, using in-place updates, and partitioning inputs into manageable chunks. These optimizations increase training throughput while lowering the memory footprint, enabling our multimodal model to scale more effectively under constrained GPU resources. The complete implementation details can be found in the attached codebase.

S2. Edge Deployment

To enable fast and memory-efficient on-device inference, we convert our PyTorch-based Vision-Language Model into an optimized NPU-executable pipeline through a sequence of conversion and hardware-specific compilation steps to run on a Windows Copilot+ Laptop[1] (Fig. F3).

1. ONNX [2] Conversion. The PyTorch model is first exported to the ONNX format using the standard PyTorch tracing pipeline. The exported ONNX graph preserves full model parameters, operator structure, and tensor formats required for downstream compiler optimization. This intermediate representation provides a hardware-agnostic bridge between the PyTorch runtime and the target NPU execution environment.

2. Weight and Activation Quantization. We then apply post-training quantization to the full ONNX model. All model weights are quantized to **8-bit integers** using a min–max calibration scheme, while activations are quantized to **16-bit** precision. Quantization statistics, the scales and offsets of the layers are computed from a representative set of **300 document samples**. The resulting quantized model runs natively on both GPU and CPU backends in PyTorch, enabling thorough validation before hardware conversion.

3. NPU Compilation. Finally, we compile the Quantized ONNX model using the Qualcomm AI Engine (QNN) compiler [3] to generate a fully NPU-executable binary. The compiler maps ONNX operators to NPU-supported kernels, performs graph-level optimizations, and produces a hardware-targeted model artifact. This step transforms the architecture into a latency-optimized, memory-efficient NPU-runnable Vision-Language model while retaining the core multimodal reasoning capabilities of the original implementation. Specifically, we use a Windows laptop equipped with a Snapdragon X Elite (X1E80100) processor featuring a 45-TOPS Hexagon-class NPU and 16 GB of unified memory.

This deployment pipeline enables our model to run efficiently on edge devices, substantially reducing memory consumption while sustaining high throughput. It provides

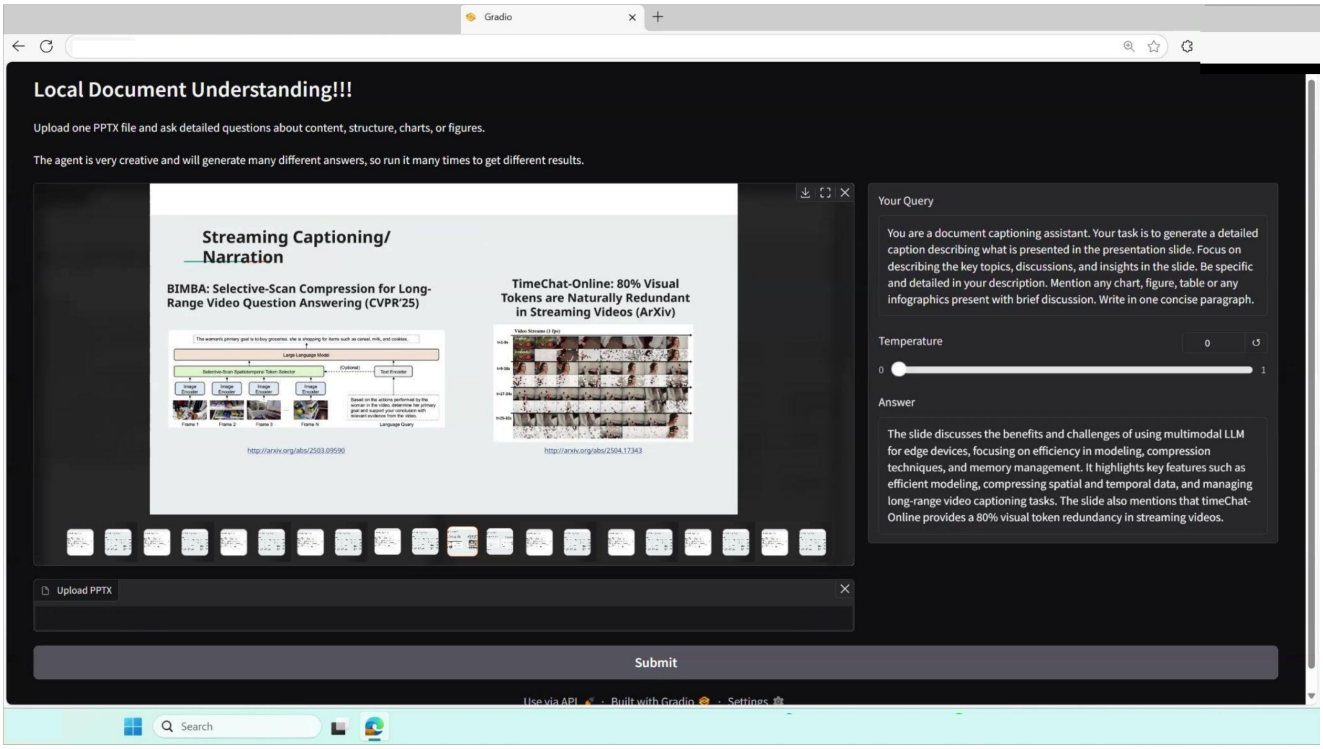


Figure F3. **Local Document Understanding on Laptop.** Screenshot of our interactive on-device system for local document understanding. Users can upload PPTX files, browse slide thumbnails, and issue natural-language queries about slide content, structure, or figures. Responses are generated entirely on-device using a Windows laptop powered by a Qualcomm Snapdragon X Elite (X1E80100) with 16 GB memory. This setup demonstrates that our pipeline performs fine-grained multimodal reasoning locally on lightweight edge hardware without relying on cloud resources. Portions of the interface have been **anonymized** using solid color blocks.

a practical path for real-world applications such as slide analysis, document assistants, and on-device multimodal agents.

S3. Additional Efficiency Analysis

Memory Efficiency. Following standard memory analyses of transformer architectures [5–7], the peak VRAM during inference can be expressed using the simple approximation:

$$\text{VRAM}_{\text{peak}} \approx \underbrace{P_{\text{B}} \times b}_{\substack{\text{parameter} \\ \text{memory}}} + \underbrace{K \times g}_{\substack{\text{KV-cache} \\ \text{(per 1k tokens)}}} + \underbrace{\mathcal{O}}_{\substack{\text{fixed} \\ \text{overhead}}} \quad (1)$$

where P_{B} is the number of parameters (in billions), b is the bytes per parameter, K is the context length measured in units of 1k tokens, g is the KV-cache cost per 1k tokens, and $\mathcal{O}$ denotes fixed activation and workspace memory. Existing document VLMs typically emit 3k–4k visual tokens per page, which leads to a steep linear increase in the token-dependent term Kg as the number of pages grows. Our model follows the same linear trend in principle; however, the crucial difference is the *slope* of this growth.

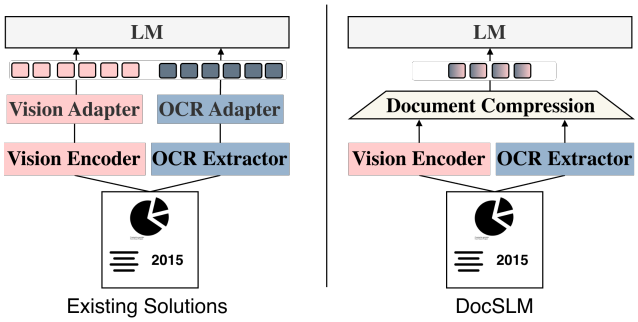


Figure F4. Prior methods process visual and OCR features independently, resulting in a large number of input tokens for the language model. In contrast, DocSLM fuses both modalities with a compression module, substantially reducing token count.

DocSLM compresses OCR, visual, and layout information into a fixed **576-token** representation per page, which dramatically reduces K for any given document. As a result, the contribution of the KV-cache and activation components in Eq. 1 grows much more slowly for our model, yielding a significantly lower overall memory footprint across long

documents compared to baselines whose vision encoders produce thousands of tokens or crops per page.

Peak GPU Memory Comparison Table T2 reports the peak GPU memory usage of several Document understanding models as the number of document pages increases from 2 to 120. All experiments were conducted using the official implementations of each model on an NVIDIA A100–80GB GPU, using the MMLongDocBench [15] dataset. We observe that existing large and medium-scale models (InternVL2-RAG[17], Docopilot[8], DocOWL2[18]) exhibit monotonic memory growth as document length increases, with memory rising sharply between 10 and 20 pages before eventually triggering out-of-memory failures. This behavior highlights the fundamental limitation of these architectures (Fig. F4) whose token counts scale linearly with the number of pages. In contrast, our streaming 2B model maintains a strictly constant peak memory footprint of 14.2 GB across all document lengths—including the 120-page setting—due to its fixed-size per-page multimodal representation and sequential stream processing. This plateau demonstrates that our design fully decouples memory usage from document length, enabling reliable, large-scale document understanding on fixed-memory hardware such as edge GPUs, laptops, and resource-constrained servers.

Model	Size	Peak Memory (GB) by Page Count					
		2	5	10	15	20	120
InternVL2-RAG [17]	8B	22.6	31.7	47.0	61.9	76.8	OOM
Docopilot[8]	8B	21.6	30.5	45.5	60.4	75.3	OOM
InternVL2-RAG [17]	**2B**	10.8	18.2	30.3	42.9	55.3	OOM
Docopilot [8]	**2B**	9.2	16.2	27.9	40.3	52.7	OOM
DocOWL2 [11]	8B	17.7	20.0	24.4	28.6	34.1	OOM
Ours	**2B**	**5.2**	**9.2**	**14.2**	**14.2**	**14.2**	**14.2**

Table T2. **Peak GPU memory usage (GB) under increasing document length.** Measurements were obtained on an NVIDIA A100–80GB GPU using the MMLongDocBench [15] dataset. Our streaming 2B model maintains a constant 14.2 GB memory footprint up to 120 pages.

Latency Vs. Accuracy Table T3 presents a detailed comparison of inference latency and accuracy on the MMLongDoc[15] benchmark across a range of state-of-the-art large multimodal models. Existing LVLMs, such as InternVL2-RAG[17] (2B/8B), Docopilot-2B[8], and VisRAG-12B[19] exhibit high computational overhead due to their large parameter counts and heavy visual token budgets (approximately 3K tokens per image). Even with retrieval-augmented pipelines (InternVL2+RAG), latency remains high (82–113 ms) and accuracy does not improve, highlighting the limitations of RAG-based pruning for long-document reasoning.

In contrast, our 2B model uses only 576 tokens per image through hierarchical multimodal compression, resulting in a 3–7× reduction in latency while simultaneously achieving the highest accuracy (22.7 Acc). This efficiency–accuracy trade-off demonstrates that compact models, when paired with structured compression and streaming mechanisms, can outperform much larger LVLMs both in speed and effectiveness, making our approach particularly suitable for real-time and edge-device deployment.

Model	Size	Tok/Image↓	Latency (ms)↓	MMLDoc (Acc↑)
InternVL2	8B	∼3,133	81.0	17.4
InternVL2+RAG	2B	∼3,133	82.9	17.2
VisRAG	12B	>3K	288.3	18.8
InternVL2	2B	∼3,133	35.9	10.5
Docopilot	2B	∼3,133	35.9	21.8
Ours	**2B**	**576**	**32.1**	**22.7**

Table T3. **Latency vs. accuracy** comparison on MMLongDoc [15] (Acc). Our 2B model achieves SOTA accuracy with substantially lower latency.

S4. Additional Ablation Studies

S4.1. Effect of OCR Confidence Threshold

To evaluate our model's robustness to OCR noise, we apply a confidence filter to OCR tokens before fusion:

$$\mathcal{T}_{\text{OCR}}(\tau) = \{\, t \in \mathcal{T} \mid \text{conf}(t) \geq \tau \},\tag{2}$$

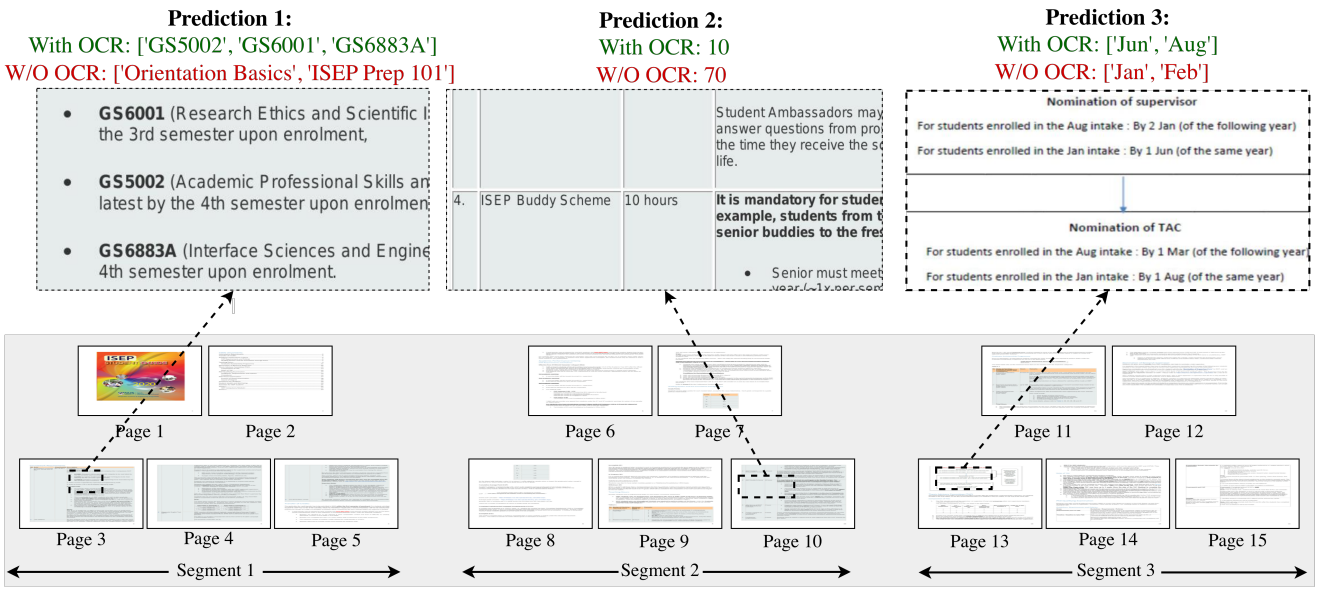
where τ is the OCR confidence threshold. Table T4 reports Mp-DocVQA accuracy for thresholds ranging from 0.0 to 0.9. Performance remains extremely stable across the full range, with the best result at $\tau = 0.0$. This indicates that our hierarchical compressor effectively absorbs OCR noise, and that aggressive filtering may remove useful but low-confidence text tokens.

OCR Threshold	0.0	0.5	0.6	0.7	0.8	0.9
Mp-DocVQA	**70.0**	69.6	69.6	69.7	69.7	69.4

Table T4. **Ablation on OCR confidence threshold.** Performance remains consistent across all thresholds, indicating that our model is robust to OCR noise and does not rely heavily on aggressive confidence filtering.

S4.2. Ablation: Effect of OCR Granularity

To study how OCR granularity influences model performance, we evaluate three configurations of the dynamic cropping pipeline, each corresponding directly to one entry in Table T5. Specifically: (i) fine-grained cropping (768–2304), which produces the largest number of crops



Query 1: Which compulsory ISEP courses does the students must have? Write the answer in list format in ascending order.
Query 2: What is the maximum hours of ISEP buddy scheme does a Singaporean ISEP students require to do?
Query 3: What is the deadline month of the January intake ISEP students need to nominate supervisors and nominations of tac? Write the answer in list format, e.g., [\"Jan\",\"Feb\"]

Figure F5. **Qualitative comparison of model predictions with and without OCR on a 15-page text-rich document.** With OCR (green), the model extracts the correct answers directly from the corresponding pages (highlighted). Without OCR (red), the model fails to recognize text-dense regions, instead hallucinating plausible-sounding but incorrect outputs. This illustrates that the failure arises from missing text perception rather than reasoning when processing visually complex document layouts.

(16–100), (ii) medium cropping (384–1536), which generates a moderate number of crops (4–49), and (iii) coarse cropping (384–1152), which yields the smallest crop count (4–18). These settings differ in the density of visual patches produced by the dynamic cropping pipeline and, accordingly, the locality of OCR tokens grounded within each patch. Table T5 reports MP-DocVQA accuracy for all three configurations. The coarse configuration (384–1152) achieves the highest accuracy, while both the medium and especially the fine-grained configurations underperform despite introducing more crops and enabling more localized OCR grounding. Higher-resolution cropping grids (e.g., 768–2304) fragment each page into many small overlapping patches, forcing OCR tokens to be split across numerous local regions.

Although this improves fine-grained text–vision alignment, it disrupts global document structure, paragraph continuity, table layout, and multi-column flow—which hinders holistic document understanding. As a result, finer-grained OCR assignments do not yield performance gains and instead degrade accuracy. In contrast, the coarse configuration (384–1152) preserves global layout while still providing adequate OCR grounding for local reasoning. This balance enables the hierarchical compressor to integrate textual cues without over-fragmenting the document.

Resolution	#Crops	MP-DocVQA
768–2304	16–100	56.7
384–1536	4–49	57.9
384–1152	4–18	**70.0**

Table T5. **Ablation on OCR granularity across dynamic crop configurations.** The #Crops column indicates the range of possible crops generated for each resized resolution; the exact number depends on the aspect ratio of the original document. Mid-range resolutions (384–1152) achieve the best balance between OCR locality and global structure.

Overall, these results show that higher OCR granularity does not necessarily improve performance. Effective long-document understanding requires a balance between local OCR grounding and global structural coherence, and the coarse 384–1152 configuration offers the most favorable trade-off.

S5. Aditional Qualitative Results

With vs without OCR Fig. F5 presents a qualitative analysis of the model's behavior on a multi-page, text-heavy academic document when OCR is present versus absent. With OCR, the model consistently retrieves correct

Figure F6. **Qualitative examples of generalization to videos.** We evaluate our model on the NewsVQA [12] benchmark, which requires understanding text embedded within video frames. We show two representative cases where our model accurately identifies the temporal segment containing the answer and correctly interprets the textual cues present in the frames. These examples highlight the model's ability to leverage multimodal signals for precise temporal localization and factually grounded answering in real video scenarios.

information from the relevant pages, demonstrating reliable grounding across segments (Pages 3, 10, and 13). In contrast, without OCR the model is unable to parse dense textual regions and instead hallucinates answers that bear no relation to the document content (e.g., inventing course names, misreading table quantities, and guessing arbitrary deadline months). These errors highlight a fundamental limitation of vision-only processing: the model fails not due to reasoning but due to its inability to perceive fine-grained text embedded in complex layouts. This underscores the necessity of OCR for long-document understanding tasks requiring precise textual extraction.

Generalization to Videos Fig. F6 presents qualitative examples illustrating our model's ability to generalize to real-world video settings. Using the NewsVQA [12] benchmark, which demands a precise understanding of text appearing within broadcast news footage, our method successfully identifies the temporal window in which the answer-relevant information is displayed. In both cases, the model tracks the textual overlays across frames, correctly localizes the segment containing the key evidence, and produces a factually accurate answer. These results demonstrate that our approach effectively leverages fine-grained textual cues in videos, enabling robust temporal grounding and reliable question answering in dynamic, text-rich video environments.

References

[1] Buy 13.8-inch surface laptop, copilot+ pc with windows - microsoft store. [Online; accessed 2025-11-21]. 2

[2] Execution providers — onnxruntime. [Online; accessed 2025-11-20]. 2

[3] Qualcomm - qnn — onnxruntime. [Online; accessed 2025-11-20]. 2

[4] Yun Dai, Vignesh Kothapalli, Qingquan Song, Shao Tang, Siyu Zhu, Steven Shimizu, Shivam Sahni, Haowen Ning, Yanning Chen, et al. Liger kernel: Efficient triton kernels for llm training. *arXiv preprint arXiv:2410.10989*, 2024. 2

[5] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023. 3

[6] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *NeurIPS*, 35:16344–16359, 2022. 1

[7] Tim Dettmers et al. Qlora: Efficient finetuning of quantized large language models. In *NeurIPS*, 2023. 3

[8] Yuchen Duan, Zhe Chen, Yusong Hu, Weiyun Wang, Shenglong Ye, Botian Shi, Lewei Lu, Qibin Hou, Tong Lu, Hongsheng Li, et al. Docopilot: Improving multimodal models for document-level understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4026–4037, 2025. 4

[9] Jianwei Feng and Dong Huang. Optimal gradient checkpoint search for arbitrary computation graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11433–11442, 2021. 1

[10] Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Chen Li, Ji Zhang, Qin Jin, Fei Huang, et al. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. *arXiv preprint arXiv:2403.12895*, 2024. 1

[11] Anwen Hu, Haiyang Xu, Liang Zhang, Jiabo Ye, Ming Yan, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. mplug-docowl2: High-resolution compressing for ocr-free multi-page document understanding. *arXiv preprint arXiv:2409.03420*, 2024. 4

[12] Soumya Jahagirdar, Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Watching the news: Towards videoqa models that can read. In *WACV*, pages 4430–4439. IEEE, 2023. 6

[13] Chenxia Li, Weiwei Liu, Ruoyu Guo, Xiaoting Yin, Kaitao Jiang, Yongkun Du, Yuning Du, Lingfeng Zhu, Baohua Lai, Xiaoguang Hu, et al. Pp-ocrv3: More attempts for the improvement of ultra lightweight ocr system. *arXiv preprint arXiv:2206.03001*, 2022. 1

[14] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *CoRR*, abs/2407.07895, 2024. 1

[15] 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. 4

[16] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 1

[17] Weiyun Wang, Shuibo Zhang, Yiming Ren, Yuchen Duan, Tiantong Li, Shuo Liu, Mengkang Hu, Zhe Chen, Kaipeng Zhang, Lewei Lu, Xizhou Zhu, Ping Luo, Yu Qiao, Jifeng Dai, Wenqi Shao, and Wenhai Wang. Needle in a multimodal haystack. *arXiv preprint arXiv:2406.07230*, 2024. 4

[18] 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*, 2023. 4

[19] Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, et al. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. *arXiv preprint arXiv:2410.10594*, 2024. 4