
This is the **published version** of the bachelor thesis:

El Aichouni Jouied, Mustapha. *Revisiting Fusion for Low-Data Multimodal Adaptation*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2025 (Grau en Intel·ligència Artificial)

This version is available at <https://ddd.uab.cat/record/336311>

under the terms of the  license.

Revisiting Fusion for Low-Data Multimodal Adaptation

Mustapha El Aichouni

June 27, 2026

Abstract

Adapting auto-regressive Vision-Language Models (VLMs) to downstream tasks is expensive, relying on massive multi-stage training over millions of paired examples and on token concatenation, which inflates the language model sequence and pays the quadratic cost of self-attention. This makes high-resolution, multi-page adaptation hard on a single GPU. In this work we ask how much performance can be squeezed from the fusion mechanism alone, with no alignment and only a small task-specific data budget. We compare three fusion families over frozen ViT and LLM backbones: Feature-wise Linear Modulation (FiLM), gated cross-attention, and token injection. The first two keep the LLM sequence length fixed, letting them exploit both backbones' inductive biases and stay trainable on a single GPU at high resolution and over many pages, whereas token injection inherits the cost and alignment dependence of concatenation. On MP-DocVQA, NewsVideoQA, and TextCaps, gated cross-attention is strongest almost everywhere, FiLM is competitive and benefits most from light fine-tuning, and token injection requires substantially more data on QA, while remaining competitive and, with an MLP projector, strongest on single-image captioning.

Keywords: Multimodal Large Language Models, Visual Fusion, Parameter-Efficient Fine-Tuning, Low-Resource Adaptation, Visual Modulation, High-Resolution Vision

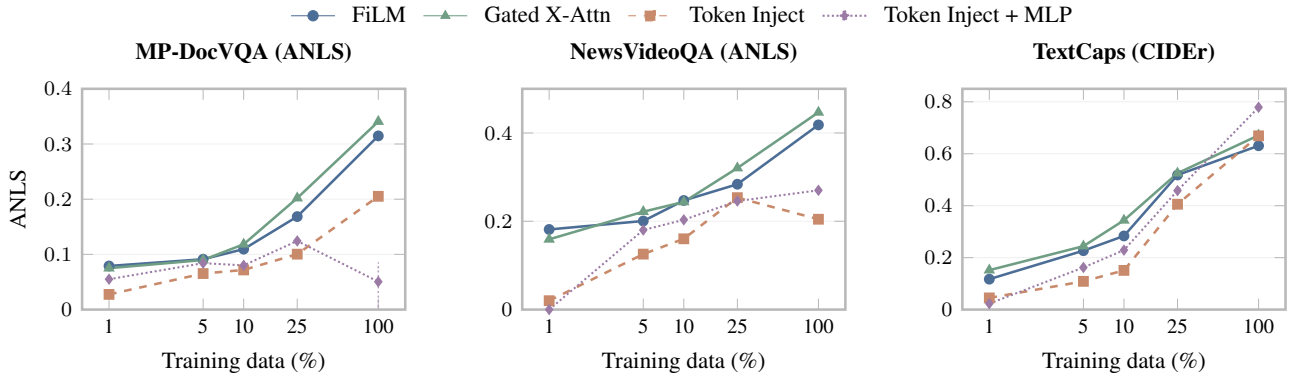


Fig. 1: **Data-efficiency curves (adapter regime, $[1, 1]$ tiles, 4 frames for NewsVideoQA), 1–100% of each dataset.** Token injection lags behind the fusion families at every budget on both QA tasks (MP-DocVQA, NewsVideoQA), even with a two-layer MLP projector, but on single-image captioning (TextCaps) it climbs steeply and overtakes them at full data, placing concatenation's weakness in the low-data QA regime rather than in concatenation itself. Error bar: min–max over three seeds (MP-DocVQA MLP, 100%); marker at mean. Other points single-run.

1 INTRODUCTION - CONTEXT OF THE PROJECT

- Contact E-mail: Mustapha.ElAichouni@uab.cat
- Supervised by: Lluís Gomez and Lei Kang (Computer Science Department)
- Academic Year 2025/26

For at least five years, training a large language model to comprehend visual cues has been an active research prob-

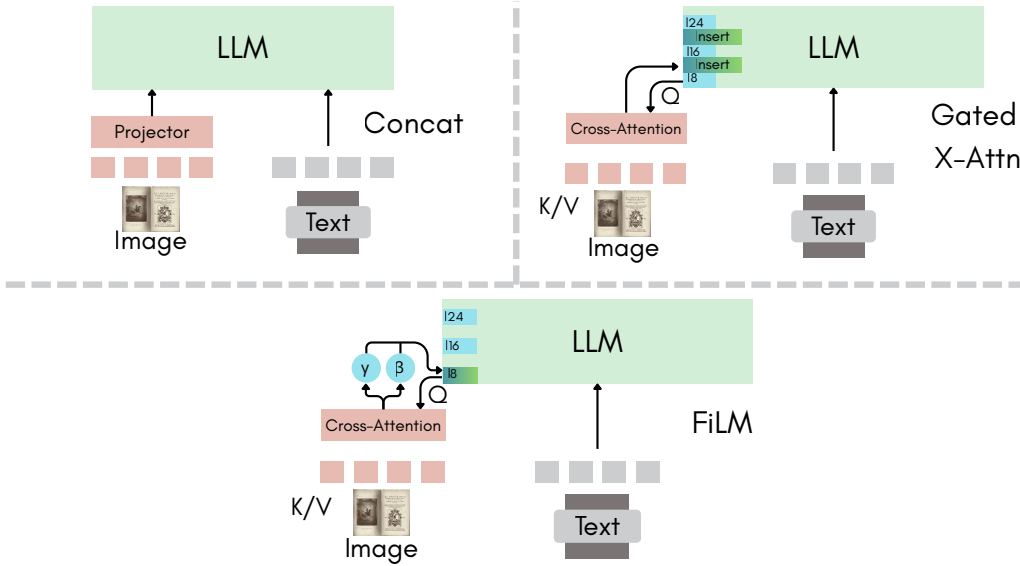


Fig. 2: **Overview of the three fusion families studied under a fixed low-data, alignment-free budget.** FiLM and gated cross-attention run a light cross-attention block between the frozen LLM hidden states and the frozen ViT features and inject the result back into the residual stream, keeping the LLM sequence length fixed. Token injection (Concat) projects visual tokens into the LLM embedding space and concatenates them, growing the sequence and paying the quadratic self-attention cost. Only the fusion module (and optionally LoRA or a fraction of the ViT) is trained.

lem with numerous architectural paradigms and frequent updates. For multimodal fusion, early systems such as Flamingo [1] employed specialized gated cross-attention layers. More recent state-of-the-art models, such as LLaVA [2] and Qwen-2 VL [3] among many others [4, 5, 6], instead use direct token concatenation.

Even while these developments demonstrate that high performance is attainable, the models frequently experience performance regressions when exposed to large quantities of visual inputs, such as document pages, and require intricate, computationally inefficient multi-stage training procedures in order to attain alignment. Beyond the quantitative impact on metrics, this problem is so computationally demanding that it hits OOM boundaries in environments with limited resources. A large part of the success of these systems is therefore inseparable from the millions of data points seen during training and from the multiple training schemes used to reach competitive performance.

Our framing is deliberately different. Instead of adding data or training stages, we ask how much performance can be squeezed from the fusion mechanism itself when adapting frozen pre-trained backbones to a specific low-data task, with no alignment whatsoever. The motivation is concrete: autoregressive MLLMs that rely on token concatenation inflate the language model sequence with visual tokens, and the quadratic cost of self-attention then makes fine-tuning on a single GPU impractical for high-resolution, multi-page inputs. We study fusion strategies that leave the LLM sequence length untouched, so that the inductive biases of both the vision encoder and the language model can be reused rather than re-learned from scratch.

To address the quadratic cost problem at retrieval time, several strategies employ techniques such as RAG or cross-encoder architectures [7]. However, this becomes problematic when heavier designs are needed to restore performance once generic retrieval fails. Recently, AVIR [8] developed

an adaptive retrieval method that uses a pre-trained Qwen 2.5-VL [3] as a question answerer without any fine-tuning and learns to provide the most relevant pages. These directions are complementary to ours: we focus on the fusion interface itself rather than on selecting which pages to read.

Our goal in this work is to give a frozen LLM the ability to comprehend long visual input sequences without a significant decline in performance, and to do it rapidly and efficiently for both training and real-world evaluation. We therefore study the following points:

- We cast visual adaptation of a frozen LLM as a choice of fusion strategy, and we provide a controlled, alignment-free, low-data comparison of three families: Feature-wise Linear Modulation (FiLM) [9], gated cross-attention, and token injection as shown in Figure 2. To our knowledge, no prior work compares these families under a fixed low-data budget for high-resolution, multi-page document and video understanding.
- We show that modulation and gated cross-attention, which keep the LLM sequence length fixed and reuse the inductive biases of the frozen ViT and LLM, adapt effectively from little data on a single GPU, whereas token injection needs substantially more data on QA, where it does not catch up even with a two-layer MLP projector, though it remains competitive on single-image captioning (Figure 1).
- We study how high-resolution tiling, multi-page context, ViT fine-tuning, and LoRA [10] shift this trade-off, and we report efficiency (parameters, GFLOPs, throughput, peak memory) alongside accuracy on MP-DocVQA [11], NewsVideoQA [12], and TextCaps [13].

2 RELATED WORK

Managing lengthy information, intricate layout, and small fonts is a significant issue for fine-grained tasks such as document understanding. Feature-wise modulation [9] functions similarly to a hypernetwork [14], conditioning a neural network in response to an external signal. Rather than producing weights, it produces a per-feature scale and shift vector, that is, a diagonal affine transformation applied channel by channel. Notably, the original FiLM work shows that this mechanism learns from little data and still generalizes, which is precisely the property we want to exploit. We also recognize its adaptability potential based on LaVi’s [15] results and several generative model applications’ accomplishments [16]. Both multimodal alignment and how these models are adapted have changed over time, and this recent wave brought significant memory overhead and training inefficiencies. Establishing a rapid adaptation technique that increases MLLM efficiency, by enabling them to handle several pages as input and stay resilient under extensive context, is the driving force behind this work.

2.1 Feature-Wise Modulation

Feature-Wise Modulation [9] is a powerful conditioning technique that applies an affine change to intermediate network representations in order to incorporate an external signal. Although Adaptive Instance Normalization [17] and generative models [16] have solidified this idea in computer vision, little is known about how effectively it can adapt LLMs to fine-grained tasks without intricate pre-training procedures.

To reduce computational overhead, recent efforts such as LaVi [15] directly inject vision-conditioned deltas into the affine parameters of LayerNorm, similar in spirit to FiLM. The requirement for two-stage training (image-caption pre-training followed by instruction adjustment), however, is a clear drawback if the goal is to adapt quickly or to significantly increase the efficiency of a pre-trained model on long-context tasks fed naively.

2.2 Multimodal Integration in Vision Language Models

For multimodal fusion, early systems such as Flamingo [1] employed dedicated gated cross-attention layers. These are efficient at inference because the visual tokens act as a read-only memory and the LLM sequence length stays fixed, but historically they were trained on very large corpora, which is exactly the alignment dependence we want to avoid. Token concatenation is a key component of more modern state-of-the-art models such as LLaVA [2] and the Qwen-2 VL [3] series. Concatenation is expressive and data-efficient at the alignment level, but it forces the language model to process long visual sequences and pays the quadratic cost of self-attention.

By using a modality pre-fusion module to compress a whole image into a single vision token, recent compression techniques such as LLaVA-Mini [18] push token reduction to its limit. Even while the model can then analyze hundreds of video frames, it still primarily uses a two-stage training paradigm and an additional pre-fusion module with signifi-

cant overhead. Closest to our low-data framing, recent work on low-resource vision-language modelling [19] combines feature modulation with token-wise gating, but it operates at a small from-scratch scale on classification-style benchmarks rather than adapting frozen large backbones to high-resolution, multi-page document understanding.

2.3 Limitations of existing approaches.

Current methods require a trade-off between accuracy and computational efficiency. Token-based concatenation frequently requires extensive pre-training and alignment due to its significant memory expense, and quadratic scaling becomes the main problem for tasks involving several images. Even while pruning and other compression techniques provide some relief, they still require some form of broad, expensive pre-training in addition to learning the compression. Although new modulation-based techniques such as LaVi show promise in terms of efficiency, they continue to resemble the LLaVA training regime and have not removed the need for pre-training. Motivated by these limitations, we treat fusion as an effective substitute for concatenation in pre-trained token-based models and study how far it can approach in-domain performance, while enabling flexibility and efficiency in extended contexts.

3 PLANNING

This section gives a summary of the project’s organization and the work plan followed throughout the development of the study.

3.1 Project Organization

During the execution of this project, an agile methodology was adopted to organize the work. Agile approaches break a project down into smaller, manageable tasks and are particularly well-suited to work marked by high variability and uncertainty, such as the development of Deep Learning models, since they allow progress to be made iteratively and leave room to adapt as difficulties arise. Among the available agile frameworks, Kanban was the one chosen for this project. In Kanban, tasks are managed on a board divided into columns that reflect their current state, typically To Do, In Progress, and Done, and each task advances across the board as it moves toward completion. This provides a clear and continuous view of progress and makes any bottlenecks easy to spot. Rather than grouping work into fixed time-boxed iterations, Kanban favours continuous flow and limits the amount of work in progress at any given moment, an approach that fits the exploratory nature of model development well, where the effort a task will demand is often hard to predict in advance. Throughout the project, progress was reviewed in weekly meetings, in which the work completed during the previous week was assessed and the priorities for the coming week were agreed upon and reordered in light of the latest experimental results.

3.2 Work Plan

The project began with a review of the literature on low-data modulation relevant to our goal. We then experimented



Fig. 3: **Representative samples from the three benchmarks.** All three require reading text in the image, but differ in task and context: question answering over a document page (a) and a video frame (b) versus text-aware captioning of a natural image (c). This QA-vs-captioning split is the axis along which the fusion families diverge in Fig. 1.

with FiLM, confirming that it performs well on lightweight captioning tasks. In parallel, we carried out a scope analysis to define our objectives and clarify what the project set out to achieve, and we produced an initial report establishing its motivation. Having settled on a clear set of goals and a structured direction, we evaluated the baseline state-of-the-art MLLMs on MP-DocVQA [11] and NewsVideoQA [12] tasks in order to measure their zero-shot performance. Next came what we refer to as the alignment experiments, in which the different methods were trained on the low-data training splits of these datasets. Alongside this, we explored a range of configurations and ran ablations, particularly on FiLM as the first method we had tested. Once the experiments were complete, we analysed the results to interpret them and draw out insights. This stage was accompanied by the writing of the progress report, followed by a proposal for the final report. Finally, a presentation is scheduled to communicate the project’s contribution. More details in the Appendix section A.3.

4 PROPOSED METHODOLOGY

4.1 Problem Definition

We study three text-centric multimodal tasks, illustrated in Figure 3. In document and video question answering (MP-DocVQA, NewsVideoQA), the model is given one or more images, scanned document pages, or frames sampled from a video, together with a natural-language question, and must return a short textual answer that is typically read directly from text in the image (a figure in a report, a caption on a news broadcast). In text-aware captioning (TextCaps), the model is given a single natural image and must produce a fluent description that incorporates text appearing in the scene. It contains roughly 28k images with about 142k captions (five per image), of which we use the standard training split under our data-budget protocol. The defining difficulty is therefore reading: the answer or caption depends on fine-grained text that is only legible at sufficient resolution.

The number of input images is a controlled variable in our setting. At evaluation we may feed only the page that contains the answer (*1-page*) or the full document of up to twenty pages (*multi-page*); for video we sample a

fixed number of frames. Adding pages or frames raises the chance the relevant content is present but lengthens the visual input, and current MLLMs degrade on MP-DocVQA and NewsVideoQA as this count grows, despite nominally supporting many images. To address this while preserving efficiency, we design fusion modules that act as a translator between visual embeddings from a multimodal encoder such as SigLIP-2 [20] and the textual hidden states of a frozen LLM. All studied fusion families share the same context generator: a light cross-attention block between the vision features of the ViT and the hidden states of the LLM at a particular layer. The families differ only in how this context is introduced.

4.2 Mathematical Definition

Let the Large Language Model (LLM) be denoted as a function ϕ . The initial token embeddings are mapped to $\mathbf{H}_0 \in \mathbb{R}^{t \times d}$, where t is the number of tokens and d is the LLM input embedding dimension. At a specific layer l , the textual hidden states are denoted as $\mathbf{H}^{(l)} \in \mathbb{R}^{t \times d_l}$, where d_l is the hidden dimension at that layer (here $d_l = d$ for the model we use).

Let $\mathcal{P} = \{\mathbf{P}_k\}_{k=1}^N$ be a set of page images, where each global image $\mathbf{P}_k \in \mathbb{R}^{H \times W \times C}$. To preserve fine-grained text readability against resolution constraints, we employ a grid-based tiling strategy [21] (see Figure 4). Given a user-defined grid parameter $[x, y]$, we define the tile width $w = \lfloor W/x \rfloor$ and height $h = \lfloor H/y \rfloor$. The image is cropped into a set of tiles $\mathbf{P}_k^{(u,v)} \in \mathbb{R}^{h \times w \times C}$, for $u \in \{0, \dots, x-1\}$ and $v \in \{0, \dots, y-1\}$.

A vision encoder γ , typically CLIP or SigLIP visual encoders, independently processes the global image and its local tiles. The resulting embeddings are concatenated to form the complete visual feature matrix:

$$\mathbf{Z}_k = [\gamma(\mathbf{P}_k) \parallel \gamma(\mathbf{P}_k^{(0,0)}) \parallel \dots \parallel \gamma(\mathbf{P}_k^{(x-1,y-1)})] \in \mathbb{R}^{N_p \times d_{img}} \quad (1)$$

where $\parallel$ denotes sequence concatenation and d_{img} is the visual hidden dimension. The encoder emits a fixed number of patch tokens per (global or local) crop; stacking the

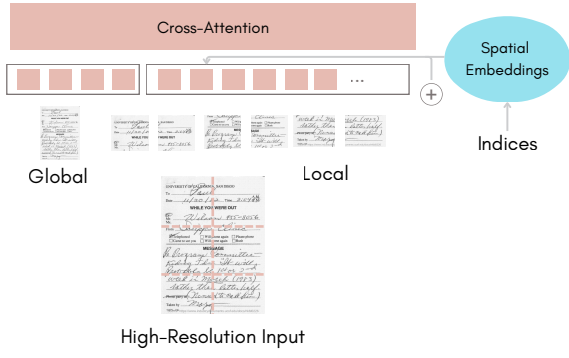


Fig. 4: **A tiling mechanism inspired by the LLaVA 1.5 setup.** The image is decomposed into a global view and multiple local crops, whose CLIP/SigLIP token embeddings are concatenated. Tile embeddings provide an explicit learned signal indicating the spatial origin of each token group (e.g., global context or specific local regions), enabling the modulator/projector to distinguish overview information from fine-grained details. The tile index embeddings are initialized with small random noise and jointly trained with the modulators.

global image and the xy tiles gives $N_p = (xy + 1)n_\gamma$ total patches, where n_γ is the per-crop token count of γ .

Shared context generator. To inject visual context into the LLM, we define a single-head cross-attention module ω that operates in a reduced bottleneck dimension b . At layer l , the text states act as queries while the visual features act as keys and values:

$$\tilde{\mathbf{H}}^{(l)} = \omega(\mathbf{Q} = \mathbf{H}^{(l)}, \mathbf{K} = \mathbf{Z}_k, \mathbf{V} = \mathbf{Z}_k) \in \mathbb{R}^{t \times b} \quad (2)$$

Running the attention in the bottleneck dimension b keeps the module light and makes the three fusion families directly comparable, since they only differ in the write-back step below.

FiLM (diagonal affine). The context vectors $\tilde{\mathbf{H}}^{(l)}$ are scaled up from b to the LLM hidden size d_l via two linear projections with weights $\mathbf{W}_\Gamma, \mathbf{W}_B \in \mathbb{R}^{b \times d_l}$, producing the modulation parameters:

$$\mathbf{\Gamma} = \tilde{\mathbf{H}}^{(l)} \mathbf{W}_\Gamma \quad \text{and} \quad \mathbf{B} = \tilde{\mathbf{H}}^{(l)} \mathbf{W}_B \quad (3)$$

We then apply Feature-wise Linear Modulation as a residual affine transformation:

$$\hat{\mathbf{H}}^{(l)} = (\mathbf{1} + \mathbf{\Gamma}) \odot \mathbf{H}^{(l)} + \mathbf{B} \quad (4)$$

where $\odot$ is the Hadamard product. By initializing $\mathbf{W}_\Gamma$ and $\mathbf{W}_B$ to zero, the parameters default to $\mathbf{\Gamma} = \mathbf{0}$ and $\mathbf{B} = \mathbf{0}$, so the operation begins as an identity mapping ($\hat{\mathbf{H}}^{(l)} = \mathbf{H}^{(l)}$) for training stability. The transformation is a per-feature scale and shift applied independently to each feature dimension of every token. It is a diagonal affine map, with no rotation, and automatic differentiation scales the text features up or down so as to boost the logit probabilities that lead to the correct answer.

Gated cross-attention. This family shares the exact context generator of FiLM and differs only in the write-back. The bottleneck context is projected back to d_l with $\mathbf{W}_O \in \mathbb{R}^{b \times d_l}$ and added residually to the hidden state through a learnable gate $\alpha^{(l)}$:

$$\hat{\mathbf{H}}^{(l)} = \mathbf{H}^{(l)} + \alpha^{(l)} (\tilde{\mathbf{H}}^{(l)} \mathbf{W}_O) \quad (5)$$

We initialize $\alpha^{(l)} = 0$ so that training also starts from the current representation. This is deliberately a stripped-down Flamingo-style block: single-head cross-attention, residual gated add, and no feed-forward network, so that FiLM and gated cross-attention remain directly comparable in capacity and only the injection mechanism (multiplicative diagonal affine versus additive residual) varies. We use single-head attention throughout: Table 8 (Appendix) shows that switching FiLM to 8-head multi-head attention reduces ANLS, and since the two families share the same context generator we expect the same to hold for gated cross-attention, so we keep both single-head for a controlled comparison.

Token injection. As a low-data instantiation of concatenation, this family projects the visual features into the LLM embedding space with a single linear layer $\mathbf{W}_{proj} \in \mathbb{R}^{d_{img} \times d}$ and prepends them to the text token embeddings $\mathbf{E}(\text{text}) \in \mathbb{R}^{t \times d}$, the LLM’s standard input embedding of the t text tokens:

$$\mathbf{H}_0 = [\mathbf{Z}_k \mathbf{W}_{proj} \parallel \mathbf{E}(\text{text})] \in \mathbb{R}^{(N_p+t) \times d} \quad (6)$$

We also evaluate a variant in which $\mathbf{W}_{proj}$ is replaced by a two-layer MLP (Linear–GELU–Linear). The single linear projection corresponds to the original LLaVA [2] recipe, while the two-layer MLP follows the improved LLaVA-1.5 [21] projector; we include both so the family is not handicapped by an under-parameterized interface.

The combined sequence is then processed by the frozen LLM self-attention. Unlike the two families above, this grows the LLM sequence to $N_p + t$ and pays the quadratic self-attention cost, which is the source of both its memory footprint and its dependence on alignment.

4.3 LLM Integration and Training Regimes

We can either use a pre-trained transformer LLM taken from an MLLM with updated weights, or one trained solely on language tokens. While we experiment with text-only models, we concentrate on LLMs from MLLMs in this work. Everything is kept frozen during training except the fusion module (Fig. 2). For FiLM and gated cross-attention, each modulated layer has its own separate module, and the input pages are shared across the chosen layers.

We employ LaVi’s [15] empirical uniform technique to select which layers receive conditioning, positioning the injection sites uniformly (early, medium, and late). For FiLM and gated cross-attention we use three layers (8, 16, 24). Token injection uses a single projection layer into the LLM space. We inject into the residual stream at each chosen layer because it performs better empirically and has a lower starting loss than the other sites we investigated. On top of the fusion module we study four adaptation regimes of increasing capacity: a plain adapter (fusion module only), LoRA on the projection layers that enter the LLM (for token injection, LoRA on every LLM layer), fine-tuning the last 25% of the ViT (the most task-specific part of the encoder, leaving the earlier, more general layers frozen), and the combination of 25% ViT fine-tuning with LoRA.

5 EXPERIMENTS

We concentrate on document and text-centric video datasets, using a three-phase approach:

- SOTA multi-image MLLMs are evaluated against trained fusion modules on MP-DocVQA, NewsVideoQA, and TextCaps.
- We measure how performance trends as additional pages and context are added, and how the fusion families compare to the SOTA baselines.
- Ablations cover backbone switching, layer placement, LoRA [10], ViT fine-tuning, resolution, and multi-page training.

5.1 Datasets

MP-DocVQA (Multi-Page Document Visual Question Answering): MP-DocVQA [11] is a large-scale dataset designed to assess the ability of models to comprehend and reason over multi-page documents. Unlike standard document VQA restricted to single-page images, it mimics real-world scenarios (reports, manuals, historical documents) where the context spans multiple pages. It comprises over 46,000 natural language questions across 6,000 documents. Answering them requires reasoning over diverse layouts, varied typography, and multi-page structure.

NewsVideoQA: NewsVideoQA [12] is a text-centric VideoQA dataset that evaluates a model’s capacity to read and reason about scene text embedded in video frames. It consists of over 8,600 question-answer pairs annotated on more than 3,000 news videos from diverse global channels. In contrast to traditional VideoQA benchmarks centered on action recognition or temporal reasoning, NewsVideoQA requires models to extract, comprehend, and aggregate textual information (news tickers, on-screen graphics, captions) across multiple frames.

TextCaps: TextCaps [13] is an image captioning dataset that requires reading and reasoning about scene text in order to produce a description. We use it both as a captioning benchmark (CIDEr and SPICE) and, in a later experiment, as a source of frozen checkpoints to study how much fine-tuning can transfer to MP-DocVQA.

5.2 Metrics

For QA we use the document-standard ANLS (Average Normalized Levenshtein Similarity) metric. For MP-DocVQA we skip predicting the correct page number, since our focus is handling many pages at once. For NewsVideoQA, since exact frames cannot be retrieved easily due to naming issues, we uniformly sample 4 frames from the video and feed them directly. These frames are much lower in resolution, so we apply a 1×1 grid over all frames and concatenate the set efficiently. For captioning on TextCaps we report CIDEr and SPICE.

The Average Normalized Levenshtein Similarity (ANLS) metric is formally defined as:

$$s(x, y) = \begin{cases} 1 - \frac{NL(x, y)}{\max(|x|, |y|)} & \text{if } 1 - \frac{NL(x, y)}{\max(|x|, |y|)} \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

For a dataset of N queries, where each query i can have multiple valid ground-truth answers, the final ANLS score is:

$$\text{ANLS} = \frac{1}{N} \sum_{i=1}^N \max_{j=1}^{M_i} s(x_i, y_{ij}) \quad (8)$$

where x_i is the predicted answer for the i -th question, y_{ij} is the j -th ground-truth answer, $NL(x, y)$ is the Levenshtein (edit) distance between strings x and y , $|x|$ and $|y|$ denote character lengths, N is the number of questions, and M_i is the number of acceptable ground-truth answers.

5.3 Implementation Details

We use Transformers, Lightning, and PyTorch for training. Since the pages in the grid-based approach are mostly vertical, similar to a DIN-A4 sheet, we use a 3×2 grid to cover all pertinent text and figure regions. Fusion is applied to the LLM residual stream at the designated layers. We use SigLIP-2 So400m as the visual encoder at 384×384 FixRes and apply our tiling strategy, and the modulated LLM is the Mistral 7B v0.1 taken from Idefics2-8B [5]. We use the Mistral 7B weights from Idefics2-8B rather than the text-only Mistral 7B v0.1 because the Idefics2 LLM has already been exposed to interleaved image-text sequences during that model’s training, so its hidden states are a more realistic substrate for visual conditioning; this isolates the effect of the fusion mechanism from the cost of teaching a purely textual LLM that visual context exists. We quantify the gap to a text-only backbone separately (Table 7 in Appendix, Qwen 2.5 swap, and the text-only entry of Table 3). To keep all backbones identical across families, every method uses these same components. The modulation strength is set to its default of 1.0. The cross-attention bottleneck dimension b is fixed at 512 across all main-table experiments for fairness, and the up-projections to d_l generate the gamma and beta parameters (we ablate $b = 1024$ separately). A single A40 GPU with 46 GB of memory is used for training and evaluation. Every model is trained for five epochs with AdamW and a learning rate of $1e-3$. FiLM and gated cross-attention run at batch size 12; token injection is too memory-expensive at that batch size, so it runs at batch size 1 with gradient accumulation 12 (effective batch 12). Every family is given the same data budget, so MP-DocVQA and TextCaps use $[1, 1]$ tiles in the main comparison.

5.4 Results

We organize results as a progression of controlled settings. We first compare all families under the low-resolution $[1, 1]$, zero-alignment budget (Table 1) and trace how each scales with training data (Figure 1). We then add a light fine-tuning stage from frozen checkpoints (Table 2), move to high-resolution $[3, 2]$ tiles (Table 3), and finally stress-test the best fusion model at deployment scale: many document pages (Table 4) and many video frames (Table 5). Each paragraph below states the exact setting it evaluates.

Main comparison ($[1, 1]$ tiles, zero-alignment). Table 1 reports the three fusion families across the four adaptation regimes on MP-DocVQA, NewsVideoQA, and TextCaps, with parameters and efficiency. Three findings stand out. First, gated cross-attention is the strongest family in almost

TABLE 1: **Fusion families compared under a fixed low-data, alignment-free budget, across four adaptation regimes.** Triplet entries (GFLOPs, it/s, Peak Mem) are ordered MP-DocVQA / NewsVideoQA / TextCaps. The visual sequence length is 729 / 2916 / 729 tokens for MP-DocVQA / NewsVideoQA / TextCaps across all families under the [1, 1] budget (NewsVideoQA uses 4 frames). FiLM and gated cross-attention use 3 uniform layers (8, 16, 24); token injection uses 1 linear projection. Best per column in **bold**. *it/s* values are means.

Regime	Method	Layers	Train Params	MP-DocVQA ANLS	NewsVideoQA ANLS	TextCaps CIDEr / SPICE	GFLOPs D / N / T	it/s D / N / T	Peak Mem (GB) D / N / T
Adapter	FiLM	3	22.4M	0.3147	0.4183	0.6309 / 0.1806	7.99 / 17.45 / 7.99	0.74 / 0.56 / 0.74	29.5 / 30.1 / 29.5
	Gated X-Attn	3	16.15M	0.3407	0.4467	0.671 / 0.1875	6.4 / 15.9 / 6.4	0.72 / 0.56 / 0.72	29.3 / 29.5 / 29.3
	Token Inject	1	4.71M	0.2052	0.2046	0.669 / 0.216	6.88 / 27.52 / 6.88	1.73 / 0.35 / 1.73	20.7 / 32.5 / 20.7
LoRA	FiLM	3	24.4M	0.3319	0.4678	0.5534 / 0.1590	8.99 / 18.45 / 8.99	0.70 / 0.54 / 0.70	31.2 / 31.8 / 31.2
	Gated X-Attn	3	18.12M	0.3578	0.4716	0.669 / 0.1810	7.4 / 16.9 / 7.4	0.69 / 0.54 / 0.69	31.0 / 31.5 / 31.0
	Token Inject	1	25.69M	0.1825	0.2193	0.3612 / 0.1995	17.62 / 38.26 / 17.62	1.31 / 0.29 / 1.31	25.4 / 45.3 / 25.4
ViT 25%	FiLM	3	113.8M	0.3250	0.4447	0.587 / 0.1692	141.17 / 550.2 / 141.17	0.69 / 0.50 / 0.69	32.8 / 40.6 / 32.8
	Gated X-Attn	3	107.6M	0.3519	0.4490	0.594 / 0.1721	139.6 / 548.6 / 139.6	0.68 / 0.49 / 0.68	32.6 / 40.44 / 32.6
	Token Inject	1	96.16M	0.3206	0.1160	0.2338 / 0.1701	140.06 / 560.27 / 140.06	1.70 / 0.35 / 1.70	22.8 / 35.2 / 22.8
ViT 25% + LoRA	FiLM	3	115.8M	0.3163	0.4986	0.547 / 0.1589	142.17 / 551.2 / 142.17	0.66 / 0.48 / 0.66	34.8 / 42.5 / 34.8
	Gated X-Attn	3	109.5M	0.3479	0.4984	0.691 / 0.1883	140.6 / 549.6 / 140.6	0.65 / 0.48 / 0.65	34.0 / 41.9 / 34.0
	Token Inject	1	117.13M	0.1521	–	0.456 / 0.1914	150.8 / – / 150.8	1.30 / – / 1.30	26.9 / OOM / 26.9

every setting: it leads MP-DocVQA in every regime and TextCaps CIDEr at the top regime, at fewer trainable parameters than FiLM, while FiLM surpasses it on NewsVideoQA only in the highest-capacity regime, and by a very small margin. Second, captioning quality degrades as we add parameters: TextCaps CIDEr for FiLM falls from 0.6309 in the adapter regime to 0.587 once 25% of the ViT is unfrozen, and a similar drop appears for gated cross-attention, which we read as overfitting and a loss of generative fluency when more capacity is trained on a small budget. The frozen-backbone regimes are therefore a better fit for low-data generation. Third, token injection is the most data-dependent family, but its weakness is task-specific (Figure 1): on the QA tasks it never becomes competitive at this budget, even with a two-layer MLP projector, and at the highest-resolution NewsVideoQA setting it runs out of memory; on single-image captioning, however, it is already competitive in the adapter regime and, with the MLP projector, reaches the best CIDEr at full data. This is consistent with input-space concatenation needing alignment to let a frozen LLM read raw projected tokens, a cost amortized only when the sequence is short and data is plentiful.

Data scaling ([1, 1] tiles, 1–100% of each split). Figure 1 traces each family as the training budget grows. On both QA tasks token injection is the lowest curve at every budget and the gap to the fusion families does not close, even with the MLP projector; on TextCaps it starts far behind but climbs steeply and, with the MLP projector, attains the best CIDEr at full data. Concatenation is therefore competitive only when the sequence is short and data is plentiful.

A note on throughput: token injection reports a higher it/s on the single-image datasets, but its effective time-to-completion is roughly four to five times that of FiLM and gated cross-attention, because the long concatenated sequence and batch-size-1 accumulation dominate the wall-clock cost. On NewsVideoQA its it/s collapses and memory rises sharply, since the visual sequence is much longer. Because *it/s* is reported per optimizer step, wall-clock time per epoch follows directly from the dataset size and the effective batch: FiLM and gated cross-attention run at batch size 12, so one epoch is roughly (training-set size/12)/(it/s), whereas token injection runs at batch size 1 with gradient accumulation 12, so its nominal it/s must be di-

vided by 12 to obtain comparable optimizer-step throughput. This is why token injection’s higher raw it/s on single-image datasets still yields a four-to-five-times longer time-to-completion.

Squeezing fine-tuning from frozen checkpoints ([1, 1], transfer to MP-DocVQA). We next ask how much performance can be recovered by starting from the frozen checkpoints trained on TextCaps with no LoRA and no ViT fine-tuning, and then fine-tuning each family on MP-DocVQA with its own LoRA (for FiLM and gated cross-attention, LoRA on the selected layers that enter the LLM; for token injection, LoRA on every LLM layer). Table 2 shows that this light regime, which loosely mirrors the alignment-then-tuning recipe used by most VLMs, boosts every family. The effect is largest for FiLM, which overtakes gated cross-attention after fine-tuning. This is informative: although FiLM is slightly weaker than gated cross-attention in direct zero-alignment training, it responds best to a small amount of subsequent tuning.

TABLE 2: **Transfer from frozen TextCaps checkpoints (no LoRA, no ViT FT) to MP-DocVQA, 1-page evaluation.** Each family is then fine-tuned with its own LoRA.

Method	Frozen	Fine-tuned	Δ
FiLM	0.3147	0.3830	+0.0683
Gated X-Attn	0.3407	0.3740	+0.0333
Token Inject	0.2052	0.3172	+0.1120

Resolution scaling ([3, 2], 1-page MP-DocVQA). Table 3 shows that moving from [1, 1] to high-resolution [3, 2] tiles produces large ANLS gains on MP-DocVQA (1-page evaluation) for the families that can use it. A text-only backbone with no visual fusion scores essentially zero (0.0001), confirming that the gains come from the visual pathway. Gated cross-attention again leads, and both ViT fine-tuning and LoRA add further improvements. Token injection is omitted here because a full [3, 2] grid spans 5187 visual tokens, which is computationally intractable once concatenated into the LLM sequence. This is the central efficiency argument for fusion that leaves the LLM sequence length untouched.

TABLE 3: **High-resolution [3, 2] MP-DocVQA, 1-page evaluation.** ViT FT denotes fine-tuning the last 25% of the encoder.

Method	ViT FT (25%)	LoRA	ANLS
Text-only backbone	—	—	0.0001
FiLM	✗	✗	0.3805
	✗	✓	0.3950
	✓	✗	0.4098
	✓	✓	0.4290
Gated X-Attn	✗	✗	0.4242
	✗	✓	0.4216
	✓	✗	0.4518
	✓	✓	0.4544

Deployment scale I: many pages ([3, 2], up to 20-page MP-DocVQA). Taking the highest-CIDEr fusion checkpoint (fine-tuned with ViT and LoRA on [3, 2] tiles) and evaluating it on MP-DocVQA at up to 20 pages, performance does not collapse. Table 4 it reaches close to SOTA: about 0.377 ANLS on the 20-page evaluation and 0.39 after a caption-model (TextCaps) fine-tune, against 0.4124 for mPLUG-Owl3 and 0.3868 for Idefics2-8B (which only evaluates the small subset of samples that fit its budget, so its number should be read with caution). On single-page evaluation the same checkpoint reaches 0.4707. For training every method we use the same backbones, including the Mistral 7B LLM with Idefics2 weights as the speaker and SigLIP-2 So400m as the encoder.

TABLE 4: **Multi-page MP-DocVQA (max 20 pages). Our entries use the MP-DocVQA gated cross-attention checkpoint with ViT and LoRA on [3, 2] tiles.** MiniCPM v2.6 only inserts the first 12 pages due to its context-length logic.

Model	ANLS	Peak Mem
Gated X-Attn (eval)	0.3770	16.7 GB
Gated X-Attn (caption FT)	0.3900	17.0 GB
mPLUG-Owl3	0.4124	21.3 GB
Idefics2-8B	0.3868 [†]	46 GB
MiniCPM v2.6	0.2280	28.1 GB
Gated X-Attn (1-page eval)	0.4707	15 GB

[†] Out of memory on the full page budget; scored only on the subset of samples that fit, so the value is an optimistic estimate.

Deployment scale II: many frames ([3, 2], NewsVideoQA frame sweep). Table 5 compares FiLM against SOTA multi-image MLLMs as the NewsVideoQA frame count grows. FiLM is the most stable family across frames: it holds around 0.33 to 0.34 ANLS from 1 to 32 frames, while Idefics2 peaks early and then collapses (it fails entirely at 32 frames), and MiniCPM and mPLUG-Owl3 remain well below FiLM throughout. This is the same robustness story as the multi-page document case, now for video frames.

TABLE 5: **NewsVideoQA frame sweep (FiLM [3,2], 3 layers) versus SOTA.** Idefics2 collapses at 32 frames. MP-DocVQA (20 max) shown for reference. GFLOPs for FiLM: 48.976.

Model	Metric	DocVQA (20)	NewsVideoQA (frames)		
			1	4	8 / 16 / 32
MiniCPM	ANLS	0.2280	0.0497	0.0533	0.1029 / 0.0759 / 0.0658
	ACC	0.1722	0.0370	0.0380	0.0760 / 0.0528 / 0.0444
Idefics2	ANLS	0.3868	0.1250	0.3150	0.2674 / 0.0236 / X
	ACC	0.0143	0.0110	0.0053	0.0042 / 0.0000 / X
mPLUG-Owl3	ANLS	0.4124	0.0725	0.1159	0.1392 / 0.1556 / 0.1613
	ACC	0.2913	0.0380	0.0813	0.0950 / 0.1172 / 0.1172
FiLM [3,2]	ANLS	0.3385	0.3016	0.3383	0.3430 / 0.3385 / 0.3377
	ACC	0.2212	0.2471	0.2809	0.2830 / 0.2798 / 0.2788

The remaining FiLM-specific ablations, covering robustness to the page count, the effect of added capacity, and component swaps, are reported in the Appendix so that the main results stay focused on the cross-family comparison.

6 CONCLUSIONS

We compared three visual fusion families, Feature-wise Linear Modulation, gated cross-attention, and token injection, under a fixed low-data, alignment-free budget on a single GPU. The picture is consistent. Fusion strategies that leave the LLM sequence length untouched, and that reuse the inductive biases of a frozen ViT and a frozen LLM, adapt well from little data, scale to high-resolution tiling, and stay robust over many pages and frames. Gated cross-attention is the strongest family overall, FiLM is competitive and responds best to a light fine-tuning stage, and token injection, which inherits the cost and alignment dependence of concatenation, is substantially more data-dependent on QA; it does not catch up even with an MLP projector and is the first family to hit memory limits at high resolution, yet remains competitive on single-image captioning, placing its weakness in the low-data QA regime rather than in concatenation itself. Adding parameters helps QA but hurts captioning, and adding layers or multi-head attention to FiLM does not help, which together suggest that the low-data regime rewards light, well-placed modulation over raw capacity.

These results read as evidence that a meaningful fraction of multimodal performance can be obtained from the fusion mechanism alone, without the millions of paired examples and the multi-stage alignment that current VLMs depend on. We see this as motivation for architectures that deliberately exploit inductive biases to work under low data, especially in resource-constrained settings where some accuracy trade-off is acceptable. The most informative next step is to add an explicit alignment pre-training stage, in the spirit of LaVi [15], and measure how much of each family’s performance comes from fine-tuning alone versus alignment-then-fine-tuning—in particular, whether it closes the QA gap token injection cannot. We leave the remaining cross-family ablations and convergence analyses as follow-ups.

ACKNOWLEDGMENTS

I would like to thank my family, especially my parents, for their constant support, sacrifice, and example, which made

this achievement possible. I also thank my supervisors, Lluís Gomez and Lei Kang, and my colleagues for their guidance and valuable insights throughout the project.

REFERENCES

- [1] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, *et al.*, “Flamingo: a visual language model for few-shot learning,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022.
- [2] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” *Advances in neural information processing systems*, vol. 36, pp. 34 892–34 916, 2023.
- [3] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, *et al.*, “Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution,” *arXiv preprint arXiv:2409.12191*, 2024.
- [4] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, *et al.*, “Qwen3-vl technical report,” *arXiv preprint arXiv:2511.21631*, 2025.
- [5] H. Laurençon, L. Tronchon, M. Cord, and V. Sanh, “What matters when building vision-language models?” *Advances in Neural Information Processing Systems*, vol. 37, pp. 87 874–87 907, 2024.
- [6] M. Deitke, C. Clark, S. Lee, R. Tripathi, Y. Yang, J. S. Park, M. Salehi, N. Muennighoff, K. Lo, L. Soldaini, *et al.*, “Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 91–104.
- [7] M. Faysse, H. Sibille, T. Wu, B. Omrani, G. Viaud, C. Hudelot, and P. Colombo, “Colpali: Efficient document retrieval with vision language models,” in *International Conference on Learning Representations*, vol. 2025, 2025, pp. 61 424–61 449.
- [8] Z. Li, Y. Li, L. Kang, D. Karatzas, and W. Ma, “Avir: Adaptive visual in-document retrieval for efficient multi-page document question answering,” in *Proceedings of the 7th ACM International Conference on Multimedia in Asia*, 2025, pp. 1–7.
- [9] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville, “FiLM: Visual reasoning with a general conditioning layer,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [10] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “LoRA: Low-rank adaptation of large language models,” in *International Conference on Learning Representations (ICLR)*, 2022.
- [11] R. Tito, D. Karatzas, and E. Valveny, “Hierarchical multimodal transformers for multipage docvqa,” *Pattern Recognition*, vol. 144, p. 109834, 2023.
- [12] S. Jahagirdar, M. Mathew, D. Karatzas, and C. Jawahar, “Watching the news: Towards videoqa models that can read,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 4441–4450.
- [13] O. Sidorov, R. Hu, M. Rohrbach, and A. Singh, “TextCaps: A dataset for image captioning with reading comprehension,” in *European Conference on Computer Vision (ECCV)*, 2020.
- [14] D. Ha, A. Dai, and Q. V. Le, “Hypernetworks,” *arXiv preprint arXiv:1609.09106*, 2016.
- [15] T. Yue, L. Guo, Y. Tang, Z. Zhao, X. Zhu, H. Huang, and J. Liu, “Lavi: Efficient large vision-language models via internal feature modulation,” *arXiv preprint arXiv:2506.16691*, 2025.
- [16] W. Peebles and S. Xie, “Scalable diffusion models with transformers,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4195–4205.
- [17] X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1501–1510.
- [18] S. Zhang, Q. Fang, Y. Yang, and Y. Feng, “Llava-mini: Efficient image and video large multimodal models with one vision token,” in *International Conference on Learning Representations*, vol. 2025, 2025, pp. 53 285–53 310.
- [19] B.-M. Ganescu, S. Salhan, A. Caines, and P. Buttery, “Looking to learn: Token-wise dynamic gating for low-resource vision-language modelling,” in *Proceedings of the BabyLM Challenge*, 2025.
- [20] M. Tschannen, A. Gritsenko, X. Wang, M. F. Naeem, I. Alabdulmohsin, N. Parthasarathy, T. Evans, L. Beyer, Y. Xia, B. Mustafa, *et al.*, “Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features,” *arXiv preprint arXiv:2502.14786*, 2025.
- [21] H. Liu, C. Li, Y. Li, and Y. J. Lee, “Improved baselines with visual instruction tuning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 26 296–26 306.

APPENDIX

A.1 FiLM Ablations

This appendix collects the FiLM-specific ablations, since FiLM is the family whose deep-dive analysis is complete in this report. They support the conclusions of the main comparison without expanding the cross-family results section.

Table 6 studies robustness to the maximum number of pages, training with a ground-truth page plus a noisy page (a random page from the same document) to encourage filtering. FiLM accepts many pages without blowing up memory, at the cost of a gradual ANLS drop, roughly 0.02 ANLS

per doubling of the page count. Using 3 uniform layers works better than injecting at a single layer, and adding ViT SFT or LoRA gives a small further gain. Idefics2-8B starts much higher at one page but degrades far more steeply as pages accumulate, which is the opposite behavior to FiLM.

TABLE 6: **Variable-page MP-DocVQA (FiLM).** Training uses a ground-truth page plus a noisy page. ViT SFT and LoRA add only minor changes.

Model	Injection Site	Max Pages	ANLS	EM
3 Layer FiLM	Uniform	1	0.3893	0.2600
		2	0.3659	0.2400
		4	0.3486	0.2292
		8	0.3426	0.2232
		16	0.3369	0.2199
		20	0.3385	0.2211
1 Layer FiLM	Middle	1	0.3421	0.2206
		2	0.3169	0.1957
2 Layer FiLM	Middle + Late	1	0.3232	0.1986
		2	0.3037	0.1851
3 Layer FiLM + SFT ViT	Uniform	1	0.3756	0.2531
		2	0.3535	0.2339
3 Layer FiLM + SFT ViT + LoRA (3 layers)	Uniform	1	0.3856	0.2549
		2	0.3592	0.2346
Idefics2-8B	None	1	0.5963	0.0210
		2	0.5059	0.0191
		4	0.3060	0.0140
		8	0.2936	0.0120

Table 7 collects the remaining FiLM ablations at 1-page evaluation. Feeding more pages during training is not helpful once we pass two pages: 1-page and 2-page training reach a similar ceiling, while 3-page and 4-page training are worse. Swapping the LLM to a smaller text-only Qwen 2.5 (1.5B) reduces performance, which we attribute to the language model’s intrinsic size rather than to whether it was trained with images. Swapping SigLIP-2 for the native Idefics2 encoder is only slightly worse. Stacking LoRA, ViT SFT, and the $b = 1024$ bottleneck on top of the [3, 2] FiLM model gives the strongest FiLM numbers (up to 0.429 ANLS), confirming the trends from the main table.

TABLE 7: **FiLM ablations, all at 1-page inference on the ground-truth page.** The last block stacks adaptation regimes on the [3, 2] FiLM model.

Description	ANLS	EM
FiLM [3,2] (1-page training, GT)	0.3805	0.2560
FiLM [3,2] (3-page training)	0.3487	0.2233
FiLM [3,2] (4-page training)	0.3359	0.2100
LLM swapped to Qwen 2.5 (1.5B)	0.3127	0.1889
Vision encoder swapped to Idefics2	0.3704	0.2568
Idefics2 encoder (4-page training)	0.3359	0.2100
Idefics2 encoder (3-page training)	0.3487	0.2233
+ LoRA (3 layers)	0.3950	0.2686
+ LoRA + ViT SFT (25%)	0.4268	0.2955
+ bottleneck 1024	0.4290	0.2994
+ ViT SFT (35%)	0.4226	0.2990

Finally, Table 8 confirms that simply adding capacity to FiLM does not help: doubling to 6 uniform layers or switch-

ing to multi-head attention both reduce ANLS while increasing GFLOPs. The light, single-head, 3-layer configuration is the right operating point for this low-data regime.

TABLE 8: **Adding layers or multi-head attention does not help FiLM (MP-DocVQA, 20 pages).**

Configuration	ANLS	ACC	GFLOPs
FiLM [3,2] (3 layers)	0.3385	0.2212	48.976
FiLM [3,2] (6 layers)	0.3021	0.1912	97.953
FiLM [3,2] (3 layers, MHA)	0.2852	0.1751	49.178

A.2 Training stability

We track validation metrics on two fixed probes during training (12 vs 128) to compare the stability of the fusion families (Fig. 5) and of the two vision encoders (Fig. 6).

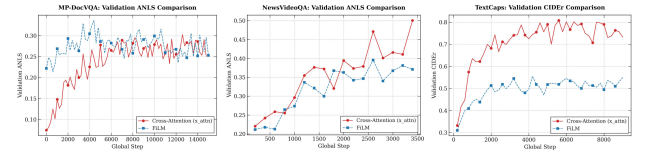


Fig. 5: **Training stability for gated cross-attention and FiLM.** Because token injection has no reliable cheap proxy and is too expensive to probe during training, we instead evaluate a fixed set of 128 validation samples every 200 steps and track how the metric evolves on this probe.

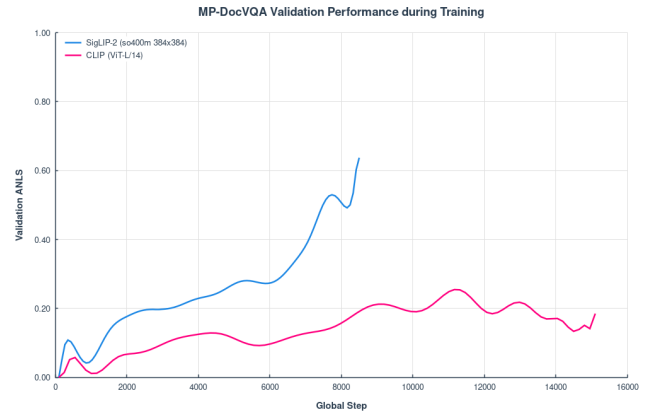


Fig. 6: **Encoder stability: CLIP vs SigLIP 2.** Validation metric tracked on a fixed 12-sample probe every 200 steps for the two vision encoders under an otherwise identical FiLM configuration.

A.3 Gantt Chart

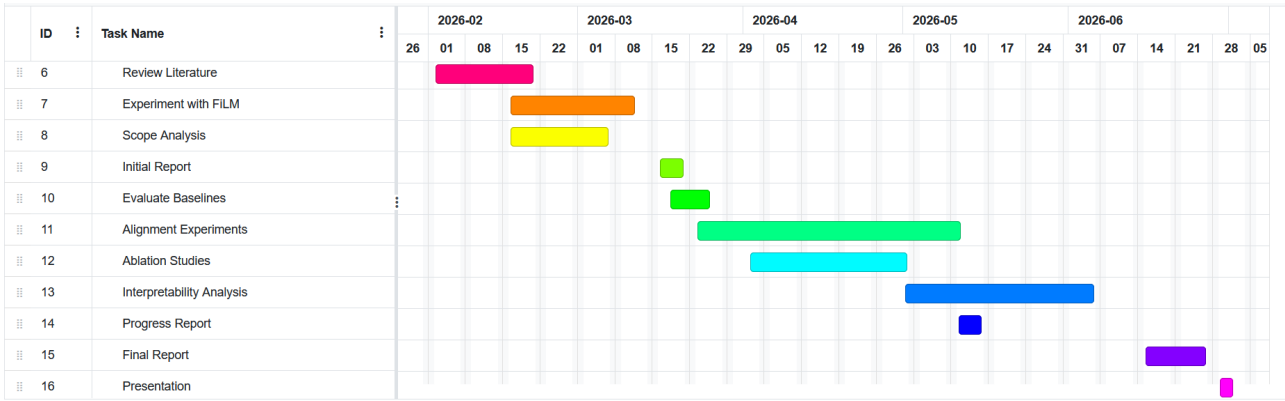


Fig. 7: A diagram of the Gantt Chart of the project.