

What Should Occupy the Context Budget? Latent Compression versus Budget-Matched Truncation in Document QA

Artem Cherepanov
HSE University
SaluteDevices
Moscow, Russia
cv4500@yandex.ru

Albina Burlova
HSE University
Moscow, Russia
aburlova@hse.ru

Abstract

Latent context compression replaces a document with a short sequence of learned vectors, but whether this beats simply keeping raw tokens under the same decoder-context budget is rarely tested. We run that test on held-out document QA, matching budgets per example against raw-token truncation and selector-specific readers trained on first- N , LLMLingua-2, and question-conditioned BM25 contexts. Our minimal compressor consists of a text encoder, learned pooling, and direct projection. It beats every tested common-cap retention baseline at nominal pooling factors $cr=8$ and $cr=16$, reaching answer perplexity 3.464 versus 5.349 for the strongest baseline at $cr=8$. The advantage persists in first-sentence free generation, and a manual check of answers is directionally consistent. Adding a post-pooling transformer hurts with twelve layers and with one layer. The twelve-layer result holds in the discovery run, two replications, and a second decoder family. Removing the post-pooling transformer also makes the compressor smaller and faster. Learned pooling followed by direct projection provides the strongest quality and efficiency among the tested compressor designs.¹

1 Introduction

Language models increasingly answer questions over documents, retrieved passages, and interaction histories. The available text often exceeds the decoder context that a serving system can afford. Each additional input position increases attention work, key-value cache memory, and latency. A serving system must therefore decide what should occupy a fixed context budget. Many document pipelines eventually allocate a fixed number of decoder positions to textual evidence. We study this allocation problem in a controlled text setting.

¹All code and evaluation artifacts are available at <https://github.com/LuLim14/CVLM-in-Doc-QA>

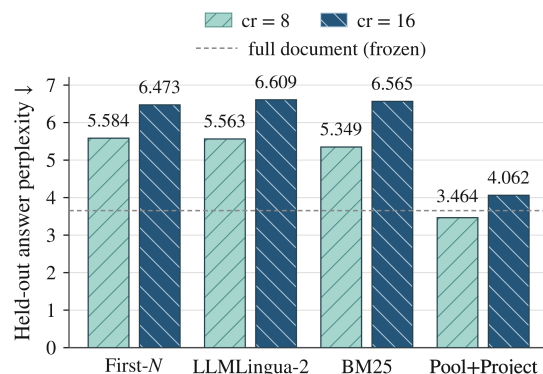


Figure 1: Under the same per-example decoder-context cap, Pool+Project outperforms the strongest tested retention baseline at $cr=8$ and $cr=16$. Each retention policy is paired with a reader trained on that policy’s outputs.

Latent context compression represents a document with a shorter sequence of continuous vectors (Mu et al., 2023; Chevalier et al., 2023; Ge et al., 2024). A learned representation can preserve evidence distributed across the document while using few decoder positions. A useful comparison gives raw tokens and learned vectors the same per-example allocation. Recent optical methods renewed interest in compression by rendering text as images (Wei et al., 2025; Cheng et al., 2026). Lee et al. (2025) found that budget-matched truncation outperformed optical compression on language modeling perplexity, while direct compression performed better on factual recall. BenchPress reports substantial variation across reading-comprehension settings and compression rates (Feldman and Artzi, 2025). We test an end-to-end latent system against task-adapted readers and stronger selectors under a shared per-example cap.

We train a query-independent compressor that combines a bidirectional text encoder, learned chunked pooling, and direct projection into the decoder space. Every system receives the same number of available document positions for each exam-

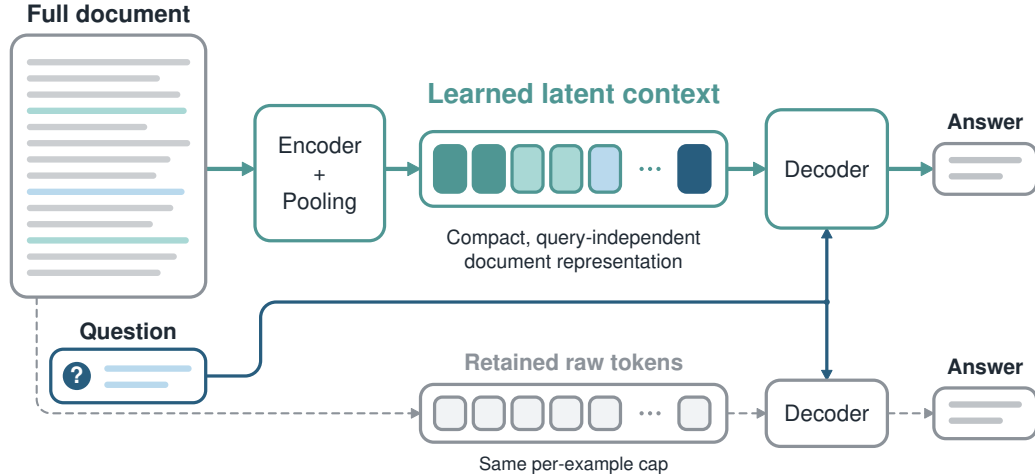


Figure 2: Learned latent context versus budget-matched truncation. The full document is compressed into a short sequence of learned latent vectors and passed to the decoder together with the question. The truncation baseline instead retains the same number of raw document tokens.

ple. The evaluation covers raw-token windows and a decoder trained on truncated inputs. LLMingua-2 (Pan et al., 2024) and BM25 each receive a reader trained on their selected contexts. We also report a frozen full-context reference. Pool+Project has the best perplexity and token accuracy at $cr=8$ and $cr=16$. Its advantage persists in length-controlled generation. Figure 1 summarizes the main comparison.

We then measure the value of processing after the pooling bottleneck. A 12-layer transformer mixer reduces held-out quality in the discovery run and two paired replications. A one-layer mixer also underperforms direct projection. A separately trained mean-pool arm shows a smaller gain from learned pooling. The same depth ordering appears with Qwen2.5 (Qwen Team, 2024).

Our contributions are:

- **A controlled common-cap QA evaluation.** Pool+Project is compared with raw-token windows and a matched-schedule first- N reader. LLMingua-2 and BM25 receive distribution-matched readers. At $cr=8$, Pool+Project reaches perplexity 3.464 against 5.349 for the strongest retention system.
- **A replicated subtractive depth ablation.** The 12-layer post-pooling mixer costs +0.63 to +0.89 held-out perplexity points in the discovery run and two paired replications. Document-cluster intervals exclude zero in every run. Qwen2.5 reproduces the depth ordering and the main compression advantage.

- **Robustness and efficiency evidence.** The depth penalty persists with fp32 optimizer masters. Removing the mixer saves 86.4M parameters. It also reduces forward latency by 5.9 to 7.4% and peak memory by 4.2% in the reported microbenchmark.

2 Related Work

Optical context compression and its critique. DeepSeek-OCR (Wei et al., 2025) compresses rendered pages roughly tenfold. Glyph (Cheng et al., 2026) and DeepSeek-OCR 2 (Wei et al., 2026) extend this direction. Side encoders render overflow context for cross-attention (Wang et al., 2024; Xing et al., 2025). Layout-aware visual memories apply the idea to agent histories (Shi et al., 2026). DeepSeek-OCR combines SAM features (Kirillov et al., 2023), convolutional downsampling, and a CLIP aggregator (Radford et al., 2021). Our initialization diagnostic places pretrained transformer blocks over latent text states after removing the image-specific interfaces. Lee et al. (2025) established the importance of budget-matched truncation and reported an advantage for direct compression on factual recall. We extend this comparison to a task-trained QA pipeline with stronger text-retention systems.

Latent compressors. Within text, contexts have been distilled into gist tokens (Mu et al., 2023) and autoencoded memory slots (Ge et al., 2024). ICAE provides the PwC QA data used in this study. Recursive compression accumulates summary vectors

over long streams (Chevalier et al., 2023). Later work studies higher rates and retrieval settings (Li et al., 2025; Cheng et al., 2024; Louis et al., 2025; Dai et al., 2025). GMSA adds one alignment layer after group pooling (Tang et al., 2026). Concurrent work scales end-to-end compression to larger training corpora and finds an MLP adapter stronger than a self-attention layer (Li et al., 2026). We study the same design axis with one and twelve post-pooling layers. The evaluation adds paired replications and a second decoder. Matched-rate studies also report strong simple pooling baselines (Feldman and Artzi, 2025; Pilchen et al., 2026).

Positioning our comparison. The closest recipes motivate a common per-example decoder-position cap and task-adapted readers. We include LLMLingua-2 (Pan et al., 2024) as a budget-calibrated learned selector. BM25 provides a question-conditioned alternative. Each selector receives its own trained reader. Complementary analyses study failure conditions for gist-style compression (Deng et al., 2025) and the information retained by compressed prompts (Łajewska et al., 2025).

3 Setup

3.1 Architecture

Figure 3 shows the full pipeline. A document of L tokens is encoded by ModernBERT-base (Warner et al., 2025) (22 blocks, hidden size 768, ~ 149 M parameters). Its hidden states are cut into non-overlapping chunks of cr encoder tokens, and each chunk is pooled by chunked attention with $K=4$ zero-initialized learnable queries. Padding is masked before the softmax, then the query outputs are averaged (Appendix A). The symmetry persists in the inspected checkpoints. All four query rows remained bitwise identical, so the realised pool is effectively single-query. The $\lceil L/cr \rceil$ latents pass an MLP projector ($768 \rightarrow 3072 \rightarrow 768$), the compressor block under study, and a second projector ($768 \rightarrow 3072 \rightarrow 2048$) into the SmolLM2-1.7B-Instruct decoder (Ben Allal et al., 2025), in the input order *question* $\rightarrow$ *latents* $\rightarrow$ *answer*. One cross-entropy loss on answer tokens trains the system end-to-end. Encoder blocks, final norm, and decoder all train. Together they account for 97.94% of parameters. Only the encoder input embeddings remain fixed. The main retention baselines therefore use task-adapted readers.

The compressor block is a ViT-Base encoder (Dosovitskiy et al., 2021) (12 blocks, width 768, ~ 86 M parameters). Its patch stem, image positions and pooler are discarded, since its input is already a latent sequence, and fresh position embeddings cap the sequence at 1,024. Four branches differ in this block alone (Table 7): it is removed (**Pool+Project**), randomly initialized, ImageNet-pretrained, or pretrained and frozen. All else is shared, so within paired campaigns each comparison isolates the post-pooling block.

3.2 Data, slices and training

We use the PwC QA data of ICAE (Ge et al., 2024). It contains Pile documents paired with GPT-4-generated questions and answers. Its test split holds 18,146 triples over 1,224 documents, about 15 questions each in contiguous blocks, so every interval in this paper is clustered by document. The 1,024-token filter passes all 18,146 items (mean document length 522.8 encoder tokens, max 723) and train/test overlap is 0 documents. The headline slice is document-disjoint from the others, while the two development slices overlap by construction. **DEV** (2,000 pairs, 135 documents) monitored development. It supplies no headline number. **FRESH** (3,000 pairs, 205 documents, none shared with DEV) was used for the final system comparison. **RQ3-DEV** (1,000 pairs, 68 documents, a subset of DEV) carries the early truncation grid and the rate-controlled reading. A 128-item probe serves the padding intervention. Absolute levels are never compared across slices.

We train for three epochs (11,325 AdamW updates, batch 64, bf16) with the curriculum $1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 16$. Moments carry across stages and checkpoints are forced at their boundaries. The discovery seed is 42. The depth contrast has two paired deterministic-sampler replications (seeds 43, 44). Two additional Pool+Project runs with those seeds form a separate system-level campaign. Exact update ranges, optimizer settings, and pairing checks are in Appendix A.

3.3 Metrics and intervals

Perplexity is the exponential of the answer-token negative log-likelihood micro-averaged over a slice, and token accuracy (TA) is teacher-forced accuracy on the same tokens. Every number is **held-out** unless labelled training loss or microbenchmark. Differences of perplexity are perplexity points and differences of log-likelihood are nats per answer to-

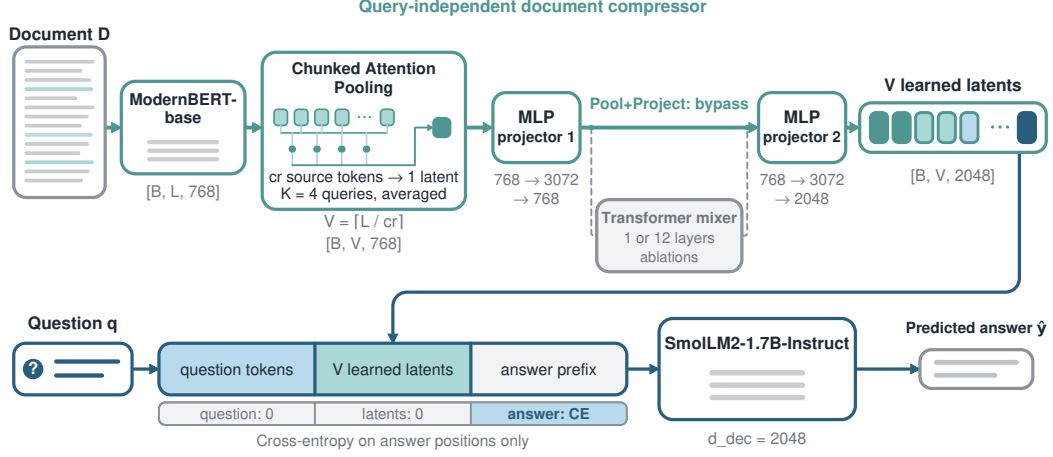


Figure 3: **Pool+Project architecture.** Source tokens are encoded by ModernBERT, compressed via Chunked Attention Pooling into $V = \lceil L/cr \rceil$ latent vectors ($V \ll L$), projected from d_{enc} to d_{dec} , and inserted after the question in the context of an autoregressive SmoLM2 decoder, which generates the answer. The single cross-entropy loss is taken on answer tokens only: the document is never reconstructed.

ken, never interchanged. Since about 15 questions share a document, the unit of resampling is the document. Unless stated otherwise, QA intervals are paired, pointwise 95% percentile-bootstrap intervals obtained by resampling documents and recomputing the token-pooled statistic over shared items. The intervals quantify held-out-document uncertainty conditional on the trained checkpoints. Row-level resampling understates width by about $1.7\times$ here, with design effects from 1.5 to 1.8, so inferential QA comparisons use document clusters. Diagnostics without intervals and the pairwise-judge percentages are descriptive. Generative metrics are secondary. Their stopping-behaviour confound and the single-rule protocol are analyzed in Appendix G.

4 Compression versus truncation at a matched budget

Budget-matched baselines. At rate cr the compressor emits $V = \lceil L/cr \rceil$ latents, so each retention policy receives the same cap of $N = V$ decoder tokens from the document. The frozen-reader controls retain the first or last N tokens, a random contiguous window, or the contiguous window with maximal reference-answer overlap. The last policy uses the gold answer and serves as a privileged window control. A separate first- N reader is trained under the compressor schedule. LLMingua-2 (Pan et al., 2024) and a question-conditioned BM25 sentence selector each receive another reader trained on that selector’s outputs. The full-context refer-

ence uses the frozen instruct decoder on the complete document.

Here cr is the encoder-side pooling factor. ModernBERT defines source length, while the decoder tokenizer measures retained text. The controlled quantity is the per-example cap on document positions presented to the decoder. Identical sample identifiers and caps are verified before paired analysis. Discrete selectors may leave a few positions unused, and their realized utilization is reported in Appendix J. Parameter count and upstream computation vary across systems. Section 5.1 reports measured costs and amortized query latency.

Compression leads the matched-budget comparison. Table 1 evaluates 3,000 held-out pairs from 205 documents. Pool+Project gives the best perplexity and token accuracy at both rates. At $cr=8$, its perplexity is 3.464, followed by 5.349 for BM25 with its matched reader. At $cr=16$, it reaches 4.062, followed by 6.473 for the first- N reader. Table 2 gives paired NLL and token accuracy differences. Every interval excludes zero. Two further Pool+Project runs reach 4.180 and 4.154 perplexity at $cr=16$.

The learned selectors remain strong controls after distribution matching. The LLMingua-2 readers reach perplexity 5.563 and 6.609 at $cr=8$ and $cr=16$. Their paired compressor-minus-baseline NLL differences are -0.4739 $[-0.4993, -0.4484]$ and -0.4865 $[-0.5120, -0.4612]$. The matched BM25 readers reach 5.349 and 6.565. The corresponding

differences are -0.4346 $[-0.4613, -0.4089]$ and -0.4799 $[-0.5035, -0.4559]$. These comparisons include selector-specific training distributions on both sides.

The main result transfers to Qwen2.5. We repeat the Pool+Project and first- N reader comparison with Qwen2.5-0.5B-Instruct (Qwen Team, 2024). Pool+Project reaches perplexity 5.395 at $cr=8$ and 6.916 at $cr=16$. The corresponding first- N readers score 13.748 and 16.958. Paired compressor-minus-reader NLL differences are -0.9354 $[-0.9854, -0.8843]$ and -0.8969 $[-0.9415, -0.8513]$. Token accuracy improves by 10.2 and 8.7 percentage points. This second decoder reproduces the matched-budget system ordering.

The full-context reference gives a metric-dependent comparison. At $cr=8$, Pool+Project has lower NLL and lower token accuracy than the frozen full-context reader. The NLL sign changes on the development slice. At $cr=16$, the full-context reader leads on both metrics. We report this row as a reference for an off-the-shelf decoder reading all available text.

Free-running answers preserve the matched-budget ordering. We score first-sentence generations on FRESH using normalized token-F1 and answer containment (Table 9). Both favor Pool+Project over the first- N reader with document-cluster intervals clear of zero. A manual check of 150 answers gives the same direction. Raw generations differ substantially in stopping behavior, so teacher-forced NLL and token accuracy remain the primary metrics.

Exploratory pairwise model judging. The PwC authors report win/lose/tie model comparisons and warn that their judge can prefer longer answers (Ge et al., 2024). We run the same comparison format with Qwen/Qwen3-8B (Qwen Team, 2025), randomized A/B order, greedy decoding, and question, reference, and candidate answers as input. Against matched-SFT, Pool+Project wins/loses/ties on 49.0/32.5/18.5% of FRESH at $cr=16$ and 51.7/35.0/13.3% at $cr=8$. Raw stopping has the same direction. Appendix H provides the prompt, parser, and results for additional pairs. We treat the judge percentages as descriptive evidence.

	$cr=8$		$cr=16$	
Decoder document context	PPL	TA	PPL	TA
<i>no compression, frozen decoder</i>				
full document (reference)	3.653 / 0.719		(same)	
<i>matched budget, trained decoder</i>				
Pool+Project (ours)	3.464	0.681	4.062	0.649
first- N matched	5.584	0.610	6.473	0.585
LLMLingua-2 [§] matched	5.563	0.608	6.609	0.580
BM25 [‡] matched	5.349	0.620	6.565	0.584
<i>matched budget, frozen instruct decoder</i>				
answer-centred [†]	6.887	0.609	8.130	0.580
first- N tokens	8.574	0.561	9.750	0.540
LLMLingua-2 [§] , frozen	8.796	0.562	10.034	0.540
BM25 [‡] , frozen	8.351	0.574	10.038	0.539

Table 1: **Latent compression against budget-matched truncation** (FRESH held-out set). Teacher-forced perplexity / token accuracy on answer tokens, best matched-budget cell in bold, budgets matched per example as a common cap (Appendix J). The symbol cr denotes the encoder-side pooling factor. Rows above the rule use readers trained on the displayed context distribution. Remaining rows use the frozen instruct decoder. The full-document reference is independent of cr . [†]uses the gold answer and falls back to first- N when its overlap search fails. Its hit rate is 0.364. [§]uses dequantized rate control under the same cap. [‡]selects sentences using the question and restores document order before reading. Paired contrasts: Table 2. Last- N and the random window: Appendix B.

Pool+Project — baseline	Δ NLL [95% CI]	Δ TA
$cr=8$		
first- N tokens	-0.9063 $[-0.9344, -0.8766]$	$+0.1198$
answer-centred [†]	-0.6871 $[-0.7149, -0.6599]$	$+0.0716$
first- N matched reader	-0.4777 $[-0.5064, -0.4496]$	$+0.0704$
LLMLingua-2 matched reader	-0.4739 $[-0.4993, -0.4484]$	$+0.0728$
BM25 matched reader	-0.4346 $[-0.4613, -0.4089]$	$+0.0609$
full-context ref.	-0.0533 $[-0.0758, -0.0304]$	-0.0380
$cr=16$		
first- N tokens	-0.8754 $[-0.9033, -0.8486]$	$+0.1090$
answer-centred [†]	-0.6937 $[-0.7188, -0.6687]$	$+0.0687$
first- N matched reader	-0.4658 $[-0.4917, -0.4394]$	$+0.0635$
LLMLingua-2 matched reader	-0.4865 $[-0.5120, -0.4612]$	$+0.0692$
BM25 matched reader	-0.4799 $[-0.5035, -0.4559]$	$+0.0653$
full-context ref.	$+0.1063$ $[+0.0780, +0.1358]$	-0.0697

Table 2: **Paired, pointwise contrasts, document-clustered** (same FRESH slice and runs as Table 1, 5,000 resamples over 205 documents). Δ NLL in nats per answer token, compressor — baseline (negative favours the compressor). Selected Δ TA interval bounds are in Appendix C. The full-context rows provide a reference. [†]see Table 1.

5 How much post-bottleneck processing does the compressor need?

The compressor that wins the budget comparison is the mixer-free, smallest variant in our study. The four branches share all components surrounding the post-pooling block. **Pool+Project** re-

moves this block. The three mixer branches restore it with random (**Pool+12L Mixer**), vision-pretrained (**Pool+12L Pretrained-ViT Mixer**) or frozen vision-pretrained (**Pool+12L Frozen-ViT Mixer**) weights. Table 7 scores their final checkpoints at the operating rate $cr=16$ on the FRESH held-out set (3,000 rows over 205 documents) under three reading conditions: the document in its original order, the same document with its non-padding source tokens randomly permuted, and no document at all.

Depth. Adding twelve transformer blocks after the pooling bottleneck makes the system worse on both held-out metrics (Table 7). The document-cluster interval excludes zero in the discovery run and in two replications (Table 8). The result identifies the post-pooling insertion point as a poor use of depth in this architecture.

The one-layer variant also underperforms by-pass. Among the measured depths $\{0, 1, 12\}$, held-out NLL worsens monotonically: a single mixer block, with the same family, initialization and schedule as the twelve, costs $+0.027$ nats over no block at $cr=16$, and twelve is worse still (intervals in Appendix F). Direct projection after pooling is the best choice among these depths. ImageNet initialization increases the gap, and freezing those weights increases it again. The initialization analysis appears in Appendix E. A representational analysis of the trained checkpoints reports 3.9% weight drift for pretrained weights and 29.5% for random initialization. The effective ranks after the block are 85 and 162. The bypass has effective rank 170. Linear CKA with the bypass output is 0.63 for pretrained weights and 0.78 for random initialization (Appendix N).

Source-order sensitivity diagnostic. Two diagnostic columns of Table 7 separate a weak compressor from a broken run. Every branch is better with a document than without one. Their sensitivity to token order differs sharply. Permuting the real source tokens raises Pool+Project perplexity by 4.41 points. The increase is only 0.13 for the pretrained-ViT branch. The lower-quality branches are also less sensitive to source order. This intervention preserves the document’s token multiset and serves as a behavioral diagnostic.

The depth result replicates. The discovery comparison and two paired deterministic-sampler replications all favour Pool+Project at $cr=16$. Point-

wise document-cluster intervals exclude zero for NLL and token accuracy in every run (Table 8). The discovery run used the original sampler. The two replications use the deterministic sampler described in Appendix A.

The depth result transfers to a second decoder. Both branches were retrained end-to-end with Qwen2.5-0.5B-Instruct in place of SmolLM2-1.7B-Instruct under the same training and evaluation protocol, with decoder-specific tokenization and interfaces, and scored on the same FRESH slice at $cr=16$. The ordering is unchanged: held-out perplexity is 6.917 for Pool+Project against 16.820 for the 12-layer mixer, a paired document-clustered penalty of $+0.8887$ nats $[+0.8462, +0.9308]$. The token-permutation diagnostic also repeats: Pool+Project loses 14.3 perplexity points when source order is destroyed, while the mixer loses 4.9. The lower-quality branch is again less sensitive to token order. Together with the matched-budget comparison in Section 4, this experiment reproduces both main orderings on a second decoder family.

Learned pooling earns its keep. Zeroing the *trained* pool queries, which reduces the pool to uniform averaging, costs held-out quality at both rates. This intervention measures the contribution of the learned weights at inference. A separately trained mean-pool arm is worse by $+0.0193$ nats at $cr=16$ in a paired comparison. Its $cr=8$ contrast is cross-seed and therefore descriptive (Appendix F). Together, the two interventions support a small local benefit from learned pooling in this setup.

5.1 Robustness checks

An inference-time padding intervention. Padding exposure provides an alternative explanation for the depth gap because the mixer attends over padded latent slots. On fixed checkpoints, toggling the key-padding mask changes PPL by at most 0.53% on a 128-item probe. For the random mixer at $cr=16$ the change is 0.05%, compared with a 15.4% Pool+12L-versus-bypass PPL gap on FRESH. A synthetic-padding probe gives similarly small point shifts (Appendices L, M). The scale of these shifts is far below the observed depth gap. At $cr=1$ the latent budget saturates, so the synthetic probe there measures lost content.

A six-rate, fixed-checkpoint diagnostic. The curriculum confounds rate with elapsed training,

so Table 5 (appendix) scores one checkpoint per branch, taken at the end of the $cr=8$ stage, at every $cr \in \{1, 2, 4, 8, 16, 32\}$, varying only the evaluation rate within each branch. Pool+Project leads in point estimates at every rate through $cr=16$, and the depth penalty is *non-monotone*, peaking at $cr=8$. Elapsed training therefore contributes to the monotone trend observed across curriculum stages. The sweep is qualitative. At $cr=32$, beyond the rates seen by this checkpoint, perplexity and token accuracy disagree. We treat this point as unstable extrapolation.

Precision control: the depth penalty persists with fp32 optimizer masters. We batch-pair a full replication in which the mixer’s master weights, gradients, and optimizer moments are fp32 while forward parameters remain bf16. The fp32 mixer is worse than its bf16 counterpart by +0.3379 nats, and the mixer-versus-bypass depth penalty remains positive in both regimes. The precision control increases the observed depth gap. Full intervals and implementation details are in Appendix K.

Systems: parameters, latency, memory. The ablation also moves three system costs in one direction (Table 6, appendix). Dropping the twelve blocks removes parameters, shortens the forward pass at both rates, and lowers peak memory. The same branch also wins held-out quality. Document latents are query-independent and can be cached across questions. At the 15 questions per document in this corpus, measured amortised latency stays within 2.3% of truncation (Appendix I).

6 Discussion

The matched-reader experiments sharpen the system comparison. First- N , LLMLingua-2, and BM25 each have a reader trained on their own context distribution. Pool+Project leads all three at both compression rates on SmolLM2. The same comparison with Qwen2.5 gives a larger NLL advantage. The gain therefore persists across two decoder families and across retention policies that range from prefix selection to question-conditioned retrieval.

The architecture study points to a simple allocation rule inside the compressor. Learned pooling provides a small benefit, while transformer depth after pooling reduces quality. The 12-layer result repeats across three SmolLM2 runs and transfers to Qwen2.5. The one-layer arm follows the same

ordering. Order permutation further shows that Pool+Project responds more strongly to source order than the lower-quality mixer branches. These results favor preserving information at the bottleneck and projecting it directly into the decoder space.

The practical comparison joins quality with measured cost. Removing the mixer saves 86.4M parameters and lowers latency and peak memory. Document latents can be cached once and reused for multiple questions. With the query density of PwC QA, amortized latency remains within 2.3% of raw-token retention. Latent compression is therefore most attractive when a document supports repeated queries under a fixed context allocation.

7 Conclusion

We asked whether a trained latent compressor is worth its context slots, giving every matched system the same per-example cap. Pool+Project achieves the strongest quality among the tested common-cap systems on SmolLM2. Its advantage holds against first- N , LLMLingua-2, and BM25 readers trained on their respective inputs. Qwen2.5 reproduces the main comparison. Among post-pooling depths $\{0, 1, 12\}$, direct projection gives the best held-out quality. The 12-layer ordering repeats in three SmolLM2 runs and with Qwen2.5. The same removal saves 86.4M parameters and reduces measured latency and memory. Effective latent compression in this setting comes from learning what to preserve at the bottleneck and presenting it directly to the decoder.

Limitations

The evaluation uses one English document-QA dataset whose source documents fit within 1,024 encoder tokens. Its questions and answers were generated by GPT-4, and the answers are usually short. The reported evidence therefore characterizes short-document QA under this data construction.

The experiments use SmolLM2-1.7B-Instruct and Qwen2.5-0.5B-Instruct. The common-cap study controls decoder positions for every example. Training costs and upstream modules remain system-specific. Teacher-forced NLL and token accuracy are primary because the systems have different generation length distributions.

Ethics Statement

We train and evaluate on the PwC QA data of ICAE (Ge et al., 2024), which pairs Pile-derived news articles with questions and answers generated by GPT-4. The corpus contains third-party copyrighted material. We cite the upstream source and distribute no source documents. The supervision is model-generated, so the metrics measure agreement with a model-written reference. The source articles can concern identifiable people, and generated answers can contain factual errors. Released artifacts therefore contain code, evaluation scripts, and aggregate metrics.

Acknowledgments

This work is an output of a research project (HSE-BR-2025-025) implemented as part of the Basic Research Program at HSE University.

References

- Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Vaibhav Srivastav, Joshua Lochner, Clémentine Fourrier, Hugo Larcher, Ben Burtenshaw, Haojun Zhao, Caleb Fahlgren, Mathieu Morlon, Agustín Piqueres Lajarín, Xuan-Son Nguyen, Cyril Zakka, and 3 others. 2025. [SmolLM2: When smol goes big — data-centric training of a fully open small language model](#). In *Second Conference on Language Modeling*.
- Jiale Cheng, Yusen Liu, Xinyu Zhang, Yulin Fei, Wenyi Hong, Ruiliang Lyu, Weihang Wang, Zhe Su, Xiaotao Gu, Xiao Liu, Yushi Bai, Jie Tang, Hongning Wang, and Minlie Huang. 2026. [Glyph: Scaling context windows via visual-text compression](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 37145–37158, San Diego, California, United States. Association for Computational Linguistics.
- Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. 2024. [xRAG: Extreme context compression for retrieval-augmented generation with one token](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 109487–109516. Curran Associates, Inc.
- Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. [Adapting language models to compress contexts](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3829–3846, Singapore. Association for Computational Linguistics.
- Yuhong Dai, Jianxun Lian, Yitian Huang, Wei Zhang, Mingyang Zhou, Mingqi Wu, Xing Xie, and Hao Liao. 2025. [Pretraining context compressor for large language models with embedding-based memory](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28715–28732, Vienna, Austria. Association for Computational Linguistics.
- Chenlong Deng, Zhisong Zhang, Kelong Mao, Shuaiyi Li, Xinting Huang, Dong Yu, and Zhicheng Dou. 2025. [A silver bullet or a compromise for full attention? a comprehensive study of gist token-based context compression](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4861–4879, Vienna, Austria. Association for Computational Linguistics.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. [An image is worth 16x16 words: Transformers for image recognition at scale](#). In *International Conference on Learning Representations (ICLR 2021)*.
- Yair Feldman and Yoav Artzi. 2025. [No mean feat: Simple, strong baselines for context compression](#). *Preprint*, arXiv:2510.20797.
- Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2024. [In-context autoencoder for context compression in a large language model](#). In *The Twelfth International Conference on Learning Representations (ICLR 2024)*.
- Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. 2023. [Segment anything](#). In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026.
- Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. [Similarity of neural network representations revisited](#). In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR.
- Weronika Łajewska, Momchil Hardalov, Laura Aina, Neha Anna John, Hang Su, and Lluís Marquez. 2025. [Understanding and improving information preservation in prompt compression for LLMs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 17520–17541, Suzhou, China. Association for Computational Linguistics.
- Ivan Yee Lee, Cheng Yang, and Taylor Berg-Kirkpatrick. 2025. [Optical context compression is just \(bad\) autoencoding](#). *Preprint*, arXiv:2512.03643.
- Ang Li, Sean McLeish, Haozhe Chen, Nimit Kalra, Zaiqian Chen, Artem Gazizov, Venkata Anoop Suhas Kumar Morisetty, Bhavya Kailkhura,

- Harshitha Menon, Zhuang Liu, Brian R. Bartoldson, Tom Goldstein, Sanae Lotfi, Micah Goldblum, and Pavel Izmailov. 2026. [End-to-end context compression at scale](#). *Preprint*, arXiv:2606.09659.
- Zongqian Li, Yixuan Su, and Nigel Collier. 2025. [500xCompressor: Generalized prompt compression for large language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25081–25091, Vienna, Austria. Association for Computational Linguistics.
- Maxime Louis, Hervé Déjean, and Stéphane Clinchant. 2025. [PISCO: Pretty simple compression for retrieval-augmented generation](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 15506–15521, Vienna, Austria. Association for Computational Linguistics.
- Jesse Mu, Xiang Lisa Li, and Noah Goodman. 2023. [Learning to compress prompts with gist tokens](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 19327–19352. Curran Associates, Inc.
- Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. [LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981, Bangkok, Thailand. Association for Computational Linguistics.
- Hippolyte Pilchen, Edouard Grave, and Patrick Pérez. 2026. [ARC-Encoder: Learning compressed text representations for large language models](#). *Transactions on Machine Learning Research*.
- Qwen Team. 2024. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Qwen Team. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.
- Olivier Roy and Martin Vetterli. 2007. [Effective rank: A measure of effective dimensionality](#). In *15th European Signal Processing Conference (EUSIPCO)*, pages 606–610.
- Yaorui Shi, Shugui Liu, Yu Yang, Wenyu Mao, Yuxin Chen, Qi Gu, Hui Su, Xunliang Cai, Xiang Wang, and An Zhang. 2026. [MemOCR: Layout-aware visual memory for efficient long-horizon reasoning](#). *Preprint*, arXiv:2601.21468.
- Jiwei Tang, Zhicheng Zhang, Shunlong Wu, Jingheng Ye, Lichen Bai, Zitai Wang, Tingwei Lu, Lin Hai, Yiming Zhao, Hai-Tao Zheng, and Hong-Gee Kim. 2026. [GMSA: Enhancing context compression via group merging and layer semantic alignment](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28690–28704, San Diego, California, United States. Association for Computational Linguistics.
- Alex Jinpeng Wang, Linjie Li, Yiqi Lin, Min Li, Lijuan Wang, and Mike Zheng Shou. 2024. [Leveraging visual tokens for extended text contexts in multi-modal learning](#). *Preprint*, arXiv:2406.02547.
- Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2025. [Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria. Association for Computational Linguistics.
- Haoran Wei, Yaofeng Sun, and Yukun Li. 2025. [DeepSeek-OCR: Contexts optical compression](#). *Preprint*, arXiv:2510.18234.
- Haoran Wei, Yaofeng Sun, and Yukun Li. 2026. [DeepSeek-OCR 2: Visual causal flow](#). *Preprint*, arXiv:2601.20552.
- Ling Xing, Alex Jinpeng Wang, Rui Yan, Xiangbo Shu, and Jinhui Tang. 2025. [Vision-centric token compression in large language model](#). In *Advances in Neural Information Processing Systems*, volume 38, pages 33080–33110. Curran Associates, Inc.

A Architecture and training details

Chunked attention pool. The encoder states of a document are cut into non-overlapping chunks of cr tokens. Writing $C_j \in \mathbb{R}^{cr \times d}$ for chunk j and $q_1, \dots, q_K \in \mathbb{R}^d$ for the learnable queries, the latent for that chunk is

$$o_j = \frac{1}{K} \sum_{k=1}^K \text{softmax} \left(\frac{C_j q_k^\top}{\sqrt{d}} \right)^\top C_j, \quad (1)$$

with padded positions masked before the softmax. We instantiate $K=4$ queries at exactly zero. Because the initialization, attention computation, and final average are permutation-symmetric, all four rows remained bitwise identical in every inspected checkpoint. Equation 1 gives the implemented form. The realised model has one distinct query vector because all four rows remain identical.

Optimization. Training uses one accelerator, bf16, gradient checkpointing, and a fused linear cross-entropy kernel for the decoder. AdamW has $(\beta_1, \beta_2)=(0.9, 0.999)$, $\epsilon=10^{-8}$, zero weight decay, and global gradient-norm clipping at 1.0. Batch size is 64 with no gradient accumulation. The learning rate warms linearly from zero to 10^{-4} for 100 updates and is then constant. There are 11,325 optimizer updates over three epochs. The internal counter begins at one and increments after each update, so the final checkpoint is indexed 11,326. The curriculum is $cr=1$ for updates 1–1,499, $cr=2$ for 1,500–2,999, $cr=4$ for 3,000–4,499, $cr=8$ for 4,500–5,999, and $cr=16$ thereafter. Optimizer moments carry across stage boundaries and a checkpoint is forced at each boundary, so branches are compared at equal update counts within a stage. Source documents are filtered at 1,024 encoder tokens. This fixed filter passes all 18,146 test items. We cap prompts at 512 and answers at 1,024 decoder tokens, and the latent sequence is capped at 1,024 positions by the position-embedding table. The truncated-SFT reference decoder is retrained under the compressor’s own schedule (three epochs, effective batch 64, constant learning rate 10^{-4} after warmup). An earlier, lighter schedule (two epochs, rate 10^{-5}) scored 5.687/6.544, so the margin over truncated-SFT is stable under the stronger schedule. The first- N , LLMingua-2, and BM25 readers are trained separately on the contexts produced by their respective selection policies. Each uses the same three-epoch schedule.

For paired replications, the epoch permutation is a deterministic function of seed and epoch, with a separate DataLoader generator. Branch-specific modules are constructed under one seed and shared projectors and positional embeddings are then re-seeded identically. Initialization hashes and the SHA-256 hash of the first batch verify pairing. The discovery seed-42 run used the previous loader implementation and is therefore always reported separately from deterministic-sampler replications 43 and 44.

The separate system-level Pool+Project runs at seeds 43 and 44 score 4.180 and 4.154 PPL at $cr=16$. Their paired contrasts use the earlier lighter-schedule SFT reader at 6.544 PPL. The differences are -0.4483 $[-0.4736, -0.4230]$ and -0.4541 $[-0.4804, -0.4281]$ nats. These checkpoints belong to a separate campaign from the depth replications in Table 8.

Evaluation. Batch 8. Evaluation covers trained compressor checkpoints, frozen and task-adapted readers on selected raw text, and a frozen instruct reader on the full document. The slices are those of §3.2. DEV has 135 documents, FRESH has 205, and RQ3-DEV has 68. The padding probe contains 128 rows from 9 documents. Beyond the slice statistics quoted in §3.2, near-duplicate document pairs (Jaccard > 0.5) number 2 of 1,224 (0.16%) and the answer string appears in the question in 0 of 18,146 items.

Logged rate quantities. Bits per source token is a constant rescaling of mean loss in this implementation. Their ratio is 0.1409 at every compression rate. The mean training ratio combines encoder and decoder tokenizers. We therefore report the measured decoder-slot cap for each item (Appendix J).

B Budget-matched truncation grid

Context given to the decoder ($cr=4$)	PPL	TA
Pool+Project latents (ours)	3.358	0.6884
answer-centred window [†]	5.816	0.6368
query-aware BM25	6.807	0.6116
first- N tokens	7.205	0.5904

Table 3: **An additional post-hoc operating rate.** The final seed-42 Pool+Project checkpoint is evaluated on FRESH at $cr=4$. teacher-forced perplexity and token accuracy are computed on answer tokens. The retention policies are read by the frozen instruct decoder at the compressor’s own per-example budget. Paired, pointwise document-clustered contrasts in nats per answer token, compressor – baseline: -0.7635 $[-0.7923, -0.7356]$ against first- N , -0.7065 $[-0.7325, -0.6798]$ against question-conditioned retrieval and -0.5493 $[-0.5773, -0.5210]$ against the privileged window, all excluding zero. [†]consults the gold answer and serves as a privileged control.

At rate cr the compressor emits $V=\lceil L/cr \rceil$ latents, and each truncation policy receives a cap of $N=V$ raw document tokens, so the configuration receives the same per-example document-context cap (§4, realized lengths in Appendix J). Table 4 adds the two policies the body omits, last- N and a random window. It is scored on the **RQ3-DEV** slice (1,000 items over 68 documents) with the compressor taken at the end of the $cr=8$ curriculum stage. Table 1 uses FRESH, so absolute levels are compared only within each table.

The **answer-centred window** is a privileged baseline. It takes the contiguous N -token span with

maximal overlap with the reference answer. Such a span is found for 36.4% of held-out items (hit rate 0.364) and the policy falls back to first- N on the rest. It provides one privileged contiguous-window control. The **full document** row is a frozen-reader reference that is independent of cr .

Context given to the decoder	PPL	TA
<i>cr=8 (matched budget)</i>		
Pool+Project latents (ours)	3.610	0.673
first- N tokens	8.094	0.572
last- N tokens	9.346	0.554
random N -token window	9.068	0.552
answer-centred window [†]	6.437	0.624
truncated-SFT decoder [‡]	5.562	0.608
<i>cr=16 (matched budget)</i>		
Pool+Project latents (ours)	5.099	0.610
first- N tokens	9.153	0.550
last- N tokens	10.370	0.537
random N -token window	10.059	0.538
answer-centred window [†]	7.615	0.593
truncated-SFT decoder [‡]	6.376	0.583
full document (reference)	3.356	0.737

Table 4: Budget-matched truncation grid, held-out, 1,000 samples over 68 documents on RQ3-DEV. Table 1 uses FRESH, so absolute levels are compared only within each table. The table gives teacher-forced PPL and token accuracy (TA) on answer tokens, best matched-budget cell per rate in bold. Free generation is reported separately and is *secondary*. ROUGE-L is 0.342 at $cr=8$ and 0.312 at $cr=16$ for the truncated-SFT row, and falls in the 0.010–0.030 band for every row scored with the frozen instruct decoder, which emits no end-of-sequence token and generates to the length cap. This band mainly reflects output length. [†]privileged answer-centred baseline (hit rate 0.364, falls back to first- N otherwise). [‡]decoder fine-tuned on documents truncated to the same per-item budget. All other rows use the frozen instruct decoder. Table 10 reports length-controlled generation.

C Paired differences for the RQ3 contrasts

The four pre-specified token-accuracy intervals promised in the caption of Table 2 (FRESH, pointwise document-clustered, 5,000 resamples) all exclude zero: at $cr=8$, $[+0.1145, +0.1249]$ against first- N , $[+0.0666, +0.0767]$ against the privileged window, $[+0.0656, +0.0752]$ against truncated-SFT, and $[-0.0429, -0.0330]$ against the full-context reference. At $cr=16$ they are $[+0.1046, +0.1137]$, $[+0.0641, +0.0733]$, $[+0.0592, +0.0679]$, and $[-0.0758, -0.0638]$ respectively.

D The rate-controlled grid

Table 5 is the rate-controlled grid read qualitatively in the robustness checks.

cr	Pool+Project	Pool+12L Mixer	+Pretr. ViT	Δ depth (PPL pts)
1	3.833	4.216	7.586	+0.383
2	3.138	3.571	7.513	+0.434
4	3.029	3.619	7.458	+0.590
8	3.610	4.308	7.453	+0.698
16	5.099	5.594	7.577	+0.496
32	7.182	7.119	7.719	-0.063

Table 5: **Rate-controlled reading: one checkpoint, six rates.** Held-out, teacher-forced perplexity on answer tokens. One checkpoint per branch from the end of the $cr=8$ curriculum stage is scored at every $cr \in \{1, 2, 4, 8, 16, 32\}$ on a single 1,000-row held-out set of 68 documents. Best per row in bold. **This is the paper’s only six-rate controlled sweep.** Within each branch, weights, data, and elapsed training are fixed while evaluation rate varies. It separates rate from the curriculum-stage confound. Δ depth = Pool+12L Mixer – Pool+Project in perplexity *points*, positive meaning the added block is worse. It is non-monotone and reaches its largest value at $cr=8$. It changes sign only in the $cr=32$ row, where all three branches have degraded to perplexity 7.1–7.7 and token accuracy near 0.56 and where token accuracy still marginally favours Pool+Project (0.568 against 0.565). Because this checkpoint had trained only to $cr=8$, its $cr=16$ and $cr=32$ rows are extrapolation beyond the range it saw. Different evaluation set from Table 7. Absolute levels are compared only within each table. One run per branch. The grid is read qualitatively and no intervals are attached.

	Params	Forward (ms)		Peak mem.
	(non-dec.)	$cr=8$	$cr=16$	$cr=8$
Pool+Project	163.2M	68.5	60.2	3.92 GB
Pool+12L Mixer	249.6M	72.8	65.0	4.09 GB
difference	-86.4M	-5.9%	-7.4%	-4.2%

Table 6: **Cost of the post-pooling block (microbenchmark).** Parameter counts cover every non-decoder module of the compressor (encoder, pooling stage, both projectors, and the post-pooling block where present). The decoder is identical in the two branches, so it is excluded from the parameter column. Timing and memory include it. *Batched teacher-forcing forward latency, batch=8, source=1024, prompt=16, answer=32, A100, bf16, excluding generation.* These are forward-pass measurements. The difference row is Pool+Project minus Pool+12L Mixer. A negative entry means Pool+Project is smaller, faster or leaner. Every entry here is a microbenchmark measurement. Held-out quality for the same two branches is in §5.

Document given	Pool+ Project	Pool+12L Mixer	+Pretr. ViT	+Frozen ViT
<i>held-out PPL (lower is better)</i>				
original order	4.062	4.688	7.504	7.782
tokens permuted	8.475	8.165	7.636	7.825
none	9.924	9.674	9.975	10.096
<i>held-out token accuracy (higher is better)</i>				
original order	0.649	0.625	0.560	0.556
tokens permuted	0.546	0.551	0.558	0.555
none	0.535	0.537	0.539	0.537

Table 7: **Depth and prior ablation with two diagnostic reading conditions.** Held-out, teacher-forced perplexity and token accuracy on answer tokens at $cr=16$, final checkpoints, on the FRESH evaluation set (3,000 rows over 205 documents). Columns are the four branches (abbreviated), rows are the reading condition fed to the compressor. “Tokens permuted” independently applies a random permutation to each item’s non-padding source tokens while keeping padding at the tail. It preserves the token multiset. This and the no-document row are diagnostics. Depth intervals: Table 8. Remaining contrasts (nats/answer token): pretrained–random +0.4705 [+0.4513, +0.4900], frozen–pretrained +0.0364 [+0.0322, +0.0407]. Shuffle penalties (token-permuted minus original-order NLL) are +0.735 / +0.555 / +0.018 / +0.005 nats for Pool+Project / 12L / Pretrained / Frozen, a $\sim 40\times$ span, every interval excluding zero.

E The ImageNet-initialization diagnostic

Optical stacks place a pretrained aggregator immediately after their compressor, and our mixer occupies the position-matched slot. Alongside the random initialization of §5 we therefore also ran the same twelve blocks warm-started from supervised ImageNet classification (google/vit-base-patch16-224), trainable and frozen. These weights were trained on images and transferred to pooled text latents. We use the transfer as an initialization diagnostic. It underperforms here: 7.504/0.560 trainable and 7.782/0.556 frozen, against 4.688/0.625 for the identical architecture initialized randomly (FRESH, $cr=16$, pretrained–random +0.4705 nats [+0.4513, +0.4900], freezing a further +0.0364 [+0.0322, +0.0407]). In this setup, an ImageNet transformer-block initialization is a weaker starting point for the text-latent mixer. The observed gap is an optimization penalty for this transferred block.

Run	Pool+Proj.	+12L Mixer	Δ PPL [95% CI]
discovery (42)	4.0618	4.6876	+0.6258 [+0.5702, +0.6795]
replication (43)	4.0605	4.7145	+0.6540 [+0.6025, +0.7046]
replication (44)	4.0574	4.9512	+0.8938 [+0.8239, +0.9639]

Table 8: **The depth contrast replicates.** Held-out perplexity at $cr=16$ on the FRESH set, and the paired difference Δ PPL = PPL(Pool+12L Mixer) – PPL(Pool+Project) in perplexity points, so a positive value means the added block is worse, with pointwise document-cluster intervals over the 205 documents. The token-accuracy differences also exclude zero and favour Pool+Project: –0.0243 [–0.0268, –0.0217], –0.0258 [–0.0281, –0.0235] and –0.0342 [–0.0368, –0.0316], reported as mixer – Pool+Project. On the additive scale the same three differences are +0.143, +0.149 and +0.199 nats per answer token. The first row is labelled the *discovery* run because it was trained under the previous sampler implementation.

F Interval bounds for the one-layer and pooling contrasts

All four branches emit the same number of latents per item at $cr=16$ and are therefore scored at an identical decoder budget. Table 5 is read qualitatively and carries no intervals by design.

The one-layer and pooling contrasts quoted qualitatively in §5 carry the following FRESH-slice document-cluster intervals, in nats per answer token. One mixer block over no block at $cr=16$ (paired): +0.0272 [+0.0231, +0.0313]. One block against twelve: –0.122 [–0.131, –0.113]. Zeroing the trained pool queries: +0.0936 [+0.0846, +0.1028] at $cr=8$ and +0.1855 [+0.1732, +0.1982] at $cr=16$. The trained-from-scratch mean-pool variant against learned pooling: +0.0193 [+0.0149, +0.0236] at $cr=16$ (paired, replication seed) and +0.030 [+0.025, +0.035] at $cr=8$ (cross-seed).

G Free generation and length control

The manual row below is a post-hoc 150-item check. Its annotation artifact records one binary acceptability judgment for each system answer against the reference (cv1m_ok and sftm_ok). One annotator produced the labels. Its document-bootstrap interval reflects item sampling conditional on those labels.

Free-generation scores are *secondary* throughout this paper. The truncated-SFT training data appends an end-of-sequence token to answers. The compressor training data omits it, so 75–82% of its generations reach the 256-token cap (about 1,172 characters against 239 for the reference) while 2–

FRESH, $cr=16$	ours	matched-SFT	Δ [95% CI]
token-F1	0.3686	0.3383	+0.0303 [+0.0230, +0.0381]
containment			+0.0642 [+0.0566, +0.0723]
manual (%)	46.7	36.0	+10.7 [−0.7, +21.4]

Table 9: **Free-running and manual evaluation** (FRESH, one stopping rule: first sentence). Paired differences with document-clustered intervals, positive favours the compressor. The manual row scores 150 items and is suggestive at this sample size. Given twice the budget, matched-SFT at $cr=8$ reaches $\Delta F1$ $[-0.008, +0.007]$ against the $cr=16$ compressor while containment still favours the compressor by +0.037. Under raw stopping the F1 sign flips (-0.197). Generation protocols: Appendix G.

3% of the SFT decoder’s do (about 246 characters). Any precision-weighted overlap score therefore measures stopping behaviour. Table 10 scores the same two systems under four protocols on the RQ3-DEV slice (68 documents). We make **first_sentence** primary because it operates independently of reference length. The two truncation protocols use oracle length. The reversal appears for Pool+Project. Against a matched-SFT decoder given twice the budget, the $cr=16$ compressor has equal F1 and higher containment.

Protocol	P+P	SFT	Δ [95% CI]
<i>length-free</i>			
first-sent. R-L F	0.3302	0.3032	+0.0269 [+0.0125, +0.0411]
first-sent. BLEU-4	12.04	9.97	+2.07 [+1.23, +2.85]
<i>raw (confounded by length)</i>			
raw R-L F	0.1351	0.3122	−0.1770 [−0.1928, −0.1614]
raw R-L recall	0.5115	0.3256	+0.1860 [+0.1727, +0.1993]
raw R-L precision	0.0852	0.3400	−0.2548
<i>oracle-length</i>			
trunc-word R-L F	0.3710	0.3259	+0.0451 [+0.0324, +0.0573]

Table 10: Free generation, held-out, on the RQ3-DEV slice (1,000 items over 68 documents). P+P = Pool+Project at $cr=16$ and SFT = the truncated-SFT decoder at $cr=16$. R-L = ROUGE-L. Δ is P+P − SFT, with document-clustered 95% intervals. The *first-sent.* block is primary: it is the only protocol here that operates independently of reference length. The *raw* block is reported to make the artifact visible: F-score favours the SFT decoder and recall favours the compressor by a comparable margin, because the compressor generates to the cap, and the *trunc-word* row truncates the prediction to the reference’s word count and is therefore *oracle-length*, which is why no claim rests on it. Exact match is 0.0000 for every configuration scored with the frozen instruct decoder, which also emits nothing on 70.8–90.6% of items, so frozen-decoder rows are excluded from generation tables and exact match is dropped entirely (§3.3). The branch ordering is identical under all four protocols and matches the perplexity ordering.

H Pairwise-judge protocol

The exploratory judge in §4 uses the Hugging Face repository identifier Qwen/Qwen3-8B through its tokenizer chat template, with `add_generation_prompt=True` and thinking disabled. Inputs contain the PwC question, the reference answer stored in the Pool+Project dump, and the two generated answers. The source document is excluded. For first-sentence evaluation, each answer is cut at the first occurrence of “.”, “!”, or “?” followed by whitespace. A Python RNG with seed 42 independently assigns Pool+Project to side A or B for each item. The exact user message is:

```
You are grading two answers to a question
about a document.
Question: {q}
Reference answer: {ref}
Answer A: {a}
Answer B: {b}
Which answer is more consistent with the
reference answer? Judge only factual
agreement with the reference, not length
or style. Reply with exactly one token:
A, B, or TIE.
```

Generation uses batches of 16, greedy decoding (`do_sample=False`), and at most four new tokens. After uppercasing and stripping the decoded continuation, outputs beginning with A or B are mapped back through the randomized order. Every other output, including an explicit TIE, is counted as a tie. There are no retries. The tie category therefore includes nonconforming output. Per-item verdicts were stored. The reported summaries have no document-clustered uncertainty and are descriptive.

Every completed pair contains all 3,000 FRESH items. Under first-sentence stopping, Pool+Project has descriptive win/lose/tie rates of 44.8/33.9/21.3% against the SmoLLM2 12-layer mixer at $cr=16$. The within-Qwen depth pair is 51.0/20.3/28.7%. For Pool+Project–matched-SFT at $cr=16$, raw stopping gives 53.0/35.0/12.0%, the same observed direction as first-sentence stopping.

I Amortised serving cost

The compressor adds a document encoder to the serving path. Its latents are query-independent, so encoding is paid once per document and reused across questions. At $cr=16$, the measured document-side compression path for a 512-token document takes 28.5 ms. A query that reuses the

cached latents then takes 83.9 ms, the same measured latency as a truncation query that re-reads raw tokens. A first, cold query therefore costs 112.4 ms, while the cost amortised over the 15 questions this corpus asks per document is 85.8 ms, 2.3% above truncation. The independently measured amortised values at $cr=8$ are 92.5 against 90.8 ms, a 1.9% overhead. The cached document representation is 0.125 MB at $cr=16$ and 0.25 MB at $cr=8$. These are microbenchmark timings on one card with synthetic inputs of fixed length, measuring prefill to the first token. They characterize the reported setup.

J Evaluation alignment

Four things a reader is entitled to check about Table 1 are separated below.

Identical items. Paired FRESH analyses are configured to require the same 3,000 sample identifiers and 205 document clusters in both per-sample dumps. Incomplete dumps cause the analysis to stop. The earlier RQ3-DEV grid contains 1,000 items over 68 documents and is reported separately. No comparison in §4 crosses evaluation sets.

Identical decoder-slot cap. For item i , all matched-budget policies receive the same document cap $V_i = \min(\lceil L_i / cr \rceil, V_{\max})$. Text selectors are retokenized before evaluation and may use fewer than V_i positions because their selection units are discrete. Every output stays within the cap. The comparison uses a matched per-example cap, while realized counts can differ.

LLMLingua-2 calibration. The budget-calibrated selector of §4 uses `llmlingua==0.2.2` with `microsoft/llmlingua-2-xlm-roberta-large-meetingbank`. Only the document is compressed. The question is excluded from selector input. We score each unique document once and bisect the continuous word-probability percentile for 32 iterations, retaining the largest attainable output within the SmolLM2-token cap. Forced tokens and consecutive-token dropping are disabled, and the renderer is required to match the package API byte-for-byte on 25 documents. No selection exceeds the cap or truncates a prefix. Mean utilization is 0.9874 at $cr=8$ and 0.9746 at $cr=16$. Exact-budget fractions are 0.6820 and 0.6211. The median shortfall is zero tokens and the ninetieth percentile is two tokens. The reader is trained on these selected contexts under the shared reader schedule.

BM25 granularity. BM25 selects sentence units

and is evaluated under an implementation rule that admits a sentence only when it fits the same cap. Selected sentences are restored to document order. The reader is trained on BM25 contexts under the shared reader schedule. Sentence-level selection can leave a small portion of the cap unused.

For the four ablation branches the budget is identical by construction: they share the pooling stage, so at $cr=16$ each emits $\lceil L/16 \rceil$ latents for the same document.

K Precision-control details

The control gives fp32 master weights, gradients, and moments to exactly 86,389,248 mixer parameters while retaining bf16 forward parameters and the legacy path elsewhere. It runs the full curriculum on replication seed 43 and is batch-paired with its bf16 counterpart. On FRESH at $cr=16$, fp32-master minus bf16 for the mixer is +0.3379 [+0.3219, +0.3535] nats. The depth penalty is +0.1491 [+0.1387, +0.1596] under bf16 and +0.4870 [+0.4633, +0.5106] under fp32 masters. These pointwise intervals resample documents. The intervention jointly changes update accumulation and optimizer-state precision. The control covers random initialization under both precision regimes. Held-out contrasts provide the reported evidence.

L Padding intervention: full numbers

The body reports that masking padded latent keys moves held-out perplexity by at most 0.53%. Table 11 gives the four signed differences behind that statement. The convention is (mask on) – (mask off) evaluated on *the same* weights, so a positive entry means the mask made perplexity slightly worse. The comparison uses the magnitude $|\Delta|$. Pool+Project is absent by construction: it has no post-pooling self-attention for a key mask to act on.

M Synthetic-latent-padding probe

The second check appends explicitly masked latent slots amounting to 25/50/75% of the pooled sequence and re-scores the same items. At $cr=16$ the padded sequence stays far below the latent cap. The observed PPL shifts range from -0.06 to -0.01 across branches and mask settings. These are small point differences without intervals.

At $cr=1$ the same intervention measures truncation at the latent cap. The number of latents is

Branch	cr	Δ PPL	Δ rel.
12L Mixer	1	-0.0087	-0.17%
12L Mixer	16	+0.0025	+0.05%
12L Pretrained-ViT Mixer	1	+0.0120	+0.14%
12L Pretrained-ViT Mixer	16	+0.0445	+0.53%

Table 11: Inference-time padding-mask intervention on a single final checkpoint per branch (128 held-out items over 9 documents, seed 42). All four magnitudes are at most 0.53%. The random-mixer change at $cr=16$ is 0.05%, versus a 15.4% held-out PPL gap between the 12-layer mixer and bypass on FRESH. The intervention provides no evidence that inference-time padding explains the magnitude of the depth gap on this probe. These point estimates characterize the inference-time intervention.

$\lceil L/cr \rceil$ with L capped at 1,024 source tokens and the latent budget capped at 1,024 slots, so at $cr=1$ the budget is exactly saturated and any lengthening of the source pushes the tail of the document past the cap, where it is silently dropped. The resulting degradation (diagnostic scale: $4.72 \rightarrow 5.76$ PPL, a difference of +1.04, at 75% appended slots) measures lost content. It is largest for Pool+Project, which has no post-pooling attention at all. The $cr=1$ row of the probe is consequently excluded from the padding argument, which rests on $cr=16$ and on the mask on/off intervention of Table 11. The saturation itself is a brittleness of the architecture at the latent-budget boundary and is listed among the limitations.

N Representational analysis of the post-pooling block

The initialization diagnostic of Appendix E shows a quality gap for ImageNet-pretrained weights. This section probes the trained checkpoints at $cr=16$ on 256 held-out documents with representational metrics summarized in Figure 4.

Weight drift. The pretrained ViT moves only 3.9% from its ImageNet initialization, versus 29.5% for the random weights, a $7.5\times$ ratio (Figure 4a). Because the initializations and parameter norms differ, this ratio is descriptive.

Effective rank and alignment. Effective rank (Roy and Vetterli, 2007) quantifies the dimensionality actually used by the representation. Given a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ with singular values $\sigma_1 \geq \dots \geq \sigma_{\min(n,d)}$, define $p_i = \sigma_i / \sum_j \sigma_j$. The effective rank is

$$\text{erank}(\mathbf{X}) = \exp\left(-\sum_i p_i \log p_i\right), \quad (2)$$

the exponential of the Shannon entropy of the normalized singular-value distribution. A matrix using all dimensions equally has $\text{erank} = \min(n, d)$. A matrix concentrated on a single direction has $\text{erank} = 1$.

The projector outputs feeding the bypass, pre-trained, and random branches have effective ranks 170, 69, and 54, respectively (Figure 4b). After the block, the latter two ranks are 85 and 162. These are branch-level descriptive measurements after separate joint end-to-end training. The pretrained branch’s rank increases across its block and remains below the random and bypass branches.

Cosine alignment between input and output of the block (Figure 4c) measures how much the post-pooling transformation rotates the representation vectors. For each latent vector $\mathbf{h}$, we compute $\cos(\mathbf{h}_{\text{in}}, \mathbf{h}_{\text{out}})$ where $\mathbf{h}_{\text{in}}$ and $\mathbf{h}_{\text{out}}$ are the input and output of the post-pooling block, and report the mean over all vectors and documents. A value of 1.0 means the block preserves direction. Lower values indicate larger rotation. The pretrained initialization rotates representations the most (0.30), the random initialization less (0.44), and bypass preserves direction (1.00).

Cross-branch similarity. We measure representational similarity with linear Centred Kernel Alignment (CKA) (Kornblith et al., 2019). Given two representation matrices $\mathbf{X} \in \mathbb{R}^{n \times d_1}$ and $\mathbf{Y} \in \mathbb{R}^{n \times d_2}$ (rows are examples, columns are features), linear CKA is

$$\text{CKA}(\mathbf{X}, \mathbf{Y}) = \frac{\|\mathbf{Y}^\top \mathbf{X}\|_F^2}{\|\mathbf{X}^\top \mathbf{X}\|_F \|\mathbf{Y}^\top \mathbf{Y}\|_F}, \quad (3)$$

where both matrices are column-centred beforehand. CKA is invariant to orthogonal transformations and isotropic scaling, making it suitable for comparing representations of different dimensionality or learned under different initializations.

At the ViT output stage (Figure 4d) the random initialization is more similar to bypass (0.78) than the pretrained initialization is (0.63). This is a descriptive cross-branch comparison.

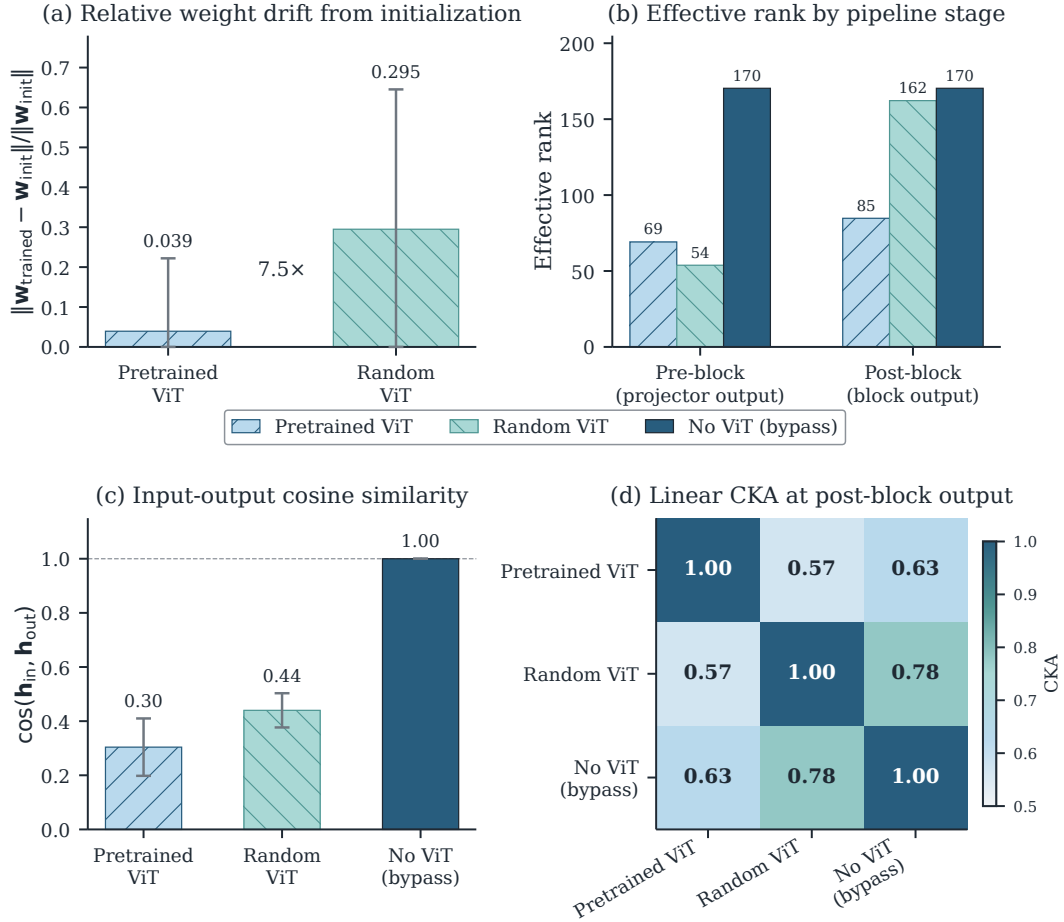


Figure 4: **Representational diagnostics for the three post-pooling branches** at $cr=16$, 256 held-out documents. (a) Mean relative weight drift from initialization, with min/max error bars for pretrained and random initializations. Bypass has no ViT weights. (b) Effective rank (Roy and Vetterli, 2007) at two pipeline stages: before the ViT block (projector output) and after. (c) Mean cosine similarity between the block’s input and output vectors, with standard-deviation error bars. Bypass is 1.0 by construction. (d) Linear CKA (Kornblith et al., 2019) similarity matrix at the ViT output stage. All diagnostics are descriptive. The error bars summarize documents within each trained checkpoint.