RANK IS THE BOTTLENECK: MEASURING AND TRAINING PLUGGABLE MEMORY FOR LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pluggable memory offers a general and portable way to provide language models with reusable context: source text is encoded once into a memory matrix that can be inserted directly into the decoder's input. Yet memory is commonly characterized by its embedding count or encoding rate, which describes the allocated budget without revealing whether those embeddings provide distinct, task-useful information. We identify rank as a key constraint on this interface. For an idealized linear attention layer, the rank of the memory matrix bounds both the feature directions that attention can read and the space of score patterns across memory positions. This perspective leads to our proposed MemTower, a two-stage framework designed to make available memory directions predictive through variable-rate continuation pretraining, explicit budget announcement, rank regularization, and fixed-rate task specialization. Across multi-passage QA and reasoning benchmarks, MemTower improves over embedding-based memory baselines; at a $\times 4$ encoding rate, its 8B configuration outperforms the best baseline on all four reasoning tasks, averaging 0.79 versus 0.72. Ablations and rank interventions further show that budget announcement and regularization improve both accuracy and the use of retained directions. MemTower preserves at least 98% of its reasoning accuracy with 16 directions, whereas removing either component requires 32 despite achieving lower full-memory accuracy. Together, these results expose a distinction overlooked by memory size alone: how much representational capacity is available, and how effectively that capacity supports prediction.

1 INTRODUCTION

In agentic systems, large language models (LLMs) often revisit repeated knowledge or documents, tool outputs and interaction histories across turns or queries. It is therefore appealing to encode such context once into a compact, reusable memory, allowing subsequent queries to consume the memory without repeatedly prefilling the full text. In this work, we study such *pluggable memory* paradigm, in which an encoder represents source text as continuous embeddings that a language decoder consumes in place of the original text (Ge et al., 2024; Cheng et al., 2024; Louis et al., 2025). We particularly focus on embedding-based memory form as the most general and portable interface, touching only the decoder's standard input embeddings, without tying to architecture-specific design. We term such encoder as the *memory tower*, analogous to the vision tower in vision–language models (Liu et al., 2023), which maps visual inputs into embeddings consumed by the decoder.

Making this interface effective requires understanding how much information the memory embeddings actually capture. Prior analyses have found high similarity among learned memory embeddings (Liu et al., 2026), echoing the anisotropy observed in language-model representations (Ethayarajh, 2019). This redundancy raises a fundamental concern: increasing the memory size may not yield a commensurate increase in the amount of information represented. The conventional compression rate measures how many source tokens are encoded into compressed embeddings, but does not reveal how effectively such capacity is utilized. A meaningful characterization of memory must therefore relate its geometric structure to the decoder's ability to exploit the representation.

We begin with the **rank of the memory matrix**, which measures the number of independent directions spanned by its embeddings. Because attention operates on combinations of these embeddings,

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

redundancy directly limits the subspace accessible to the decoder. We formalize this constraint under an idealized linear attention interface: the memory rank bounds both the space of possible attention outputs and the dimension of the attention scores over memory positions (Proposition 1). Thus, a large number of embeddings may still expose only a low-dimensional subspace to attention, making rank a structural bottleneck that memory size alone fails to capture.

The rank bound constrains also motivates towards a rank-preserving training: available directions must carry content that supports downstream inference. We thus propose **MemTower**, a two-stage framework in which a memory tower and decoder are first pretrained jointly with a general LLM objective to predict continuation across variable memory budgets, and then specialized to downstream tasks at a fixed rate. Variable-rate pretraining encourages retention of essential content under tight budgets and use of additional directions as the budget grows. We also introduce an explicit *budget announcement*, enabling the tower to allocate information across the entire memory matrix conditioned on the available budget. Lastly, a *rank regularizer* discourages representational variation from concentrating in a few directions, leading to higher rank spreading. Task specialization in the end adapts memory encoding and decoding to the target tasks and operating budget.

We evaluate five Wikipedia-based multi-passage QA benchmarks and four mathematics and knowledge reasoning benchmarks. QA scores already approach full-context performance; though, we identify certain train–evaluation overlap that could complicate interpretation (Appendix A.6). The reasoning tasks require preserving precise quantities, relations, and constraints from complete problem statements. At $\times 4$ rate, with an 8B tower and decoder, MemTower surpasses the best baseline on all four reasoning benchmarks, averaging 0.79 accuracy versus 0.72. Our ablation studies and rank analysis confirm that these design choices improve both task performance and the effective use of available memory directions. Budget announcement is particularly important, while rank regularization becomes more beneficial under tighter memory budgets. Training-time spectral measurements further show that regularization effectively broadens variation in both tower states and memory matrices. In summary, our contributions are:

- **Rank as a memory bottleneck:** We show that memory-matrix rank bounds the independent directions available through a linear attention interface. We further introduce usable rank to measure how many principal directions are needed to preserve downstream accuracy (§3).
- **Training memory to use its available rank:** We propose MemTower, a two-stage framework that combines variable-rate continuation pretraining, explicit memory-budget announcement, and spectral regularization before fixed-rate task specialization (§4).
- **Evidence across QA and reasoning:** Experiments across multi-passage QA and reasoning tasks demonstrate the effectiveness of MemTower. Component ablations and accuracy–rank analysis further connect its gains to better use of the available memory directions, while distinguishing spectral spread from task utility (§5–6).

2 Related Work

Continuous context compression. Our methodology aligns with prior works that compress context into learned continuous representations within different modules. Gist tokens (Mu et al., 2023), ICAE (Ge et al., 2024), AutoCompressor (Chevalier et al., 2023), xRAG (Cheng et al., 2024), 500xCompressor (Li et al., 2025), LLoCO (Tan et al., 2024), COCOM (Rau et al., 2025), and PISCO (Louis et al., 2025) provide context-derived representations through an embedding or soft-prompt interface. This extends soft prompting (Lester et al., 2021) by making the representation condition on context. Activation Beacon (Zhang et al., 2025) and CEPE (Yen et al., 2024) instead use internal attention states; gated cross-attention provides another access path (Alayrac et al., 2022). Apart from continuous representation, prior works have also investigated orthogonal directions such as discrete prompt compression that hard-prunes context tokens (Jiang et al., 2023; Xu et al., 2024). This work focuses on the most portable form of embedding-based context representation towards a universal interface, which is free from model-specific architectural design and ready for deployment directly.

Connections to vision–language models. Our methodology of textual memory tower resembles the visual tower in VLM design such as LLaVA (Liu et al., 2023; 2024), Qwen2-VL (Wang et al., 2024) which connect visual representation to language decoders through input embeddings. Apart from the visual modality, xRAG (Cheng et al., 2024) explicitly treats text embeddings for retrieval

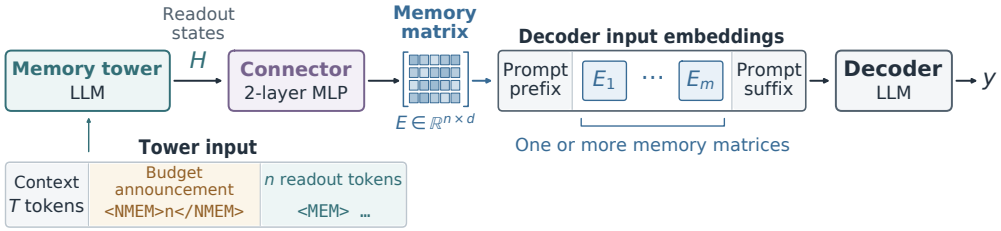


Figure 1: The pluggable memory interface in MemTower. Each chunk's T context tokens are followed by a budget announcement and n <MEM> readout tokens. The connector maps their states H to a memory matrix E. One or more such matrices ($E_1, \ldots, E_m$, $m \geq 1$) can be inserted in replacement of the original text embeddings through the decoder's standard input interface.

as a standalone modality, feeding a projection of text embeddings obtained from a frozen encoder to the decoding process. Work on visual-token redundancy (Chen et al., 2024; Cai et al., 2025) also motivates us distinguishing memory matrix size from representation utility, where our intervention focuses on the memory matrix rank instead.

Representation geometry and rank. The geometry of memory representation can be degraded by anisotropy (Gao et al., 2019; Ethayarajh, 2019): hidden states occupy a narrow cone in embedding space so that a few rogue dimensions dominate (Timkey & van Schijndel, 2021), which motivates us examining redundancy among memory embeddings. VICReg (Bardes et al., 2022), Barlow Twins (Zbontar et al., 2021), MCR^2 (Yu et al., 2020), and Seq-VCR (Arefin et al., 2025) encourage spread or reduce redundancy through geometric objectives in latent space. Our rank regularizer belongs to this broader family; we utilize a log-determinant regularization motivated upon our measurement-guided analysis of memory embeddings. Prior works have proposed effective rank (Roy & Vetterli, 2007), stable rank (Rudelson & Vershynin, 2007), and RankMe (Garrido et al., 2023) to summarize spectral concentration only. We instead highlight Usable Rank in our method that bridges the representation geometry along with the downstream decoding performance.

3 The Memory Tower: Interface, Rank, and Measurement

In this section, we first introduce the setup of memory tower and its pluggable interface with the decoder. We then show how the rank of the memory matrix constrains attention in Transformers, and define *usable rank* to measure how many leading memory directions are needed to preserve downstream accuracy, which motivates a surrogate objective for training.

3.1 The Pluggable Memory Interface

Our model pairs an LLM-based **memory tower** for context encoding with a separate LLM as the decoder that accepts the encoded memory representation. The memory tower encodes context into continuous embeddings, which replace the corresponding source tokens in the decoder's input embedding sequence alongside the remaining decoder prompt. The decoder generates from a sequence comprising one or many memory matrices, allowing contextual information to flow through the memory interface. Such paradigm achieves the *pluggable* aspect: memory representations of arbitrary knowledge, documents, or contextual information can be fed into the decoder through its standard embedding interface, without requiring any modifications to the decoder's internal architecture.

Concretely, for a chunk of T context tokens, the tower appends n <MEM> readout tokens and collects their final-layer hidden states in $H \in \mathbb{R}^{n \times d_t}$, where d_t is the tower's hidden dimension. A two-layer MLP *connector* with a SiLU nonlinearity maps each H to the decoder's d-dimensional embedding space, producing the **memory matrix** $E \in \mathbb{R}^{n \times d}$. We call n the **memory budget** and T/n the **encoding rate**, corresponding to the compression rate reported in prior work. Long contexts are encoded chunk by chunk either independently or with an overlapping window, and their memory matrices are thereby concatenated.

The decoder along with the memory tower are trained jointly: they are finetuned to learn the generation of target sequences. Further training details are provided in §4 and Appendix A.2.

Scope of the interface. This work adopts the memory form of input embeddings, which provides a general, portable interface for context representation across different LLM architectures (Mu et al., 2023; Ge et al., 2024; Louis et al., 2025): the decoder needs only to accept memory embeddings in its token embedding space. This supports direct use of existing LLM inference engines such as vLLM (Kwon et al., 2023) and reuse of cached memory matrices across queries.

3.2 Rank bounds the linear attention

The anisotropy and token redundancy discussed in §2 suggest that the number of memory embeddings alone may overstate their independent variation. We therefore examine $r := \mathrm{rank}(E)$, the number of independent directions spanned by the memory matrix, equivalently, the number of nonzero singular values. We show that this rank bounds both the feature directions that an attention head can read from memory and the attention patterns across memory positions.

In Transformers layers, an attention head reads memory by weighted-averaging memory embeddings. Let $W_Q, W_K \in \mathbb{R}^{d\times d_k}$ and $W_V \in \mathbb{R}^{d\times d_v}$ be its query, key, and value projections. We analyze these linear operations before position-dependent transformations or additive masks, absorbing the constant score scaling into the projections.

Proposition 1 (Rank bounds memory outputs and scores)**.** *For a memory matrix E of rank r:*

(a) ***Reading from memory.*** *For any attention weights $\alpha \in \mathbb{R}^n$, the weighted memory output $W_V^\top E^\top \alpha$ lies in a fixed subspace of dimension at most r.*

(b) ***Scoring memory positions.*** *As the query $q \in \mathbb{R}^d$ varies, the scores before softmax, $\ell(q) = EW_K W_Q^\top q$, span a subspace of dimension at most r.*

Appendix A.1 provides the detailed derivation and the additional rank constraint imposed by the connector. The result gives rank a concrete role at the memory interface: it bounds the independent directions available to attention, a property that embedding count alone does not capture. The significant role of rank motivates the measuring of how many memory directions the decoder actually consumes to preserve downstream performance, leading to our measurement of *usable rank* in §3.3.

3.3 Measuring usable rank

The linear rank bound leaves open how much memory variation a nonlinear decoder needs for prediction. We introduce **usable rank** through a controlled intervention: replace each memory matrix with $E^{(k)}$, retaining its leading k principal directions, and measure accuracy $a(k)$ with the decoder, memory length, and positions fixed. Usable rank is the smallest tested k whose accuracy is within relative tolerance ϵ of a_{full}, the accuracy with the original memory matrices:

$$R^\star \;=\; \min\big\{\, k :\; a(k) \;\geq\; (1-\epsilon)\, a_{\mathrm{full}} \,\big\}. \tag{1}$$

For each E, we average its rows to obtain the mean memory embedding μ, then subtract μ and apply singular value decomposition (SVD). Retaining the k principal directions accounting for the most variation and restoring μ gives

$$\mu = \tfrac{1}{n}\mathbf{1}^\top E, \qquad E - \mathbf{1}\mu = USV^\top, \qquad E^{(k)} = \mathbf{1}\mu + U_k S_k V_k^\top. \tag{2}$$

Here U_k, V_k contain the first k columns of U, V, and S_k is diagonal with the k largest singular values. Thus k bounds the number of retained directions around the mean (*centered rank*), while the average embedding is preserved. Appendix A.7 gives boundary cases and evaluation details.

A spectral surrogate for training. Usable rank is for downstream evaluation and provides no direct training gradient. Motivated to encourage rank spread, we propose a differentiable log-determinant **spectral capacity surrogate** $C(E) = \frac{1}{2}\sum_i \log(1 + \lambda_i/\alpha)$, where λ_i are the squared singular values of the centered memory matrix and $\alpha > 0$ sets the spectral scale. The logarithm's diminishing returns favor spreading variation across directions over strengthening dominant ones. A normalized form of this surrogate is adopted as auxiliary training loss in §4.1.

4 MODEL AND TRAINING METHOD

We next introduce our design of a two-stage training procedure that first teaches the memory tower and decoder to communicate through memory matrices, then adapts them to downstream tasks.

4.1 STAGE 1: BOTTLENECK PRETRAINING ON GENERAL CORPORA

Continuation as the primary objective. Standard LLM training predicts next token from the preceding text with a cross-entropy loss. We adapt this objective to memory-conditioned continuation, following prior work (Chevalier et al., 2023; Ge et al., 2024; Qian et al., 2025). Each document is split into an optional prefix span, a span encoded by the memory tower, and the remaining continuation. The decoder receives the prefix and the resulting memory embeddings, then predicts the continuation autoregressively. Cross-entropy is applied only to continuation tokens, training the tower and decoder jointly through the memory interface. This objective encourages the memory to retain semantics useful for future prediction, matching its downstream role as conditioning information.

Variable-rate exposure. To train the memory tower to adapt its information retention to different memory budgets, we sample the encoding rate log-uniformly between a minimum and maximum rate, empirically set to 4 and 64 respectively, and derive each chunk's memory budget from its length and the sampled rate. Such balanced exposure across budget scales during training encourages the tower to preserve essential context within a small memory budget while exploiting additional capacity when available, potentially shaping how information is distributed across the memory matrix. Appendix A.3 details the easy-to-hard curriculum for the rate scheduling.

Conditioning on the memory budget. Variable-rate training raises a structural issue: with causal attention, each <MEM> readout can see preceding tokens but not how many readouts follow. For a given context, a smaller-budget memory is therefore a prefix of the larger-budget memory. Without an explicit budget announcement, variable-rate training naturally induces a Matryoshka-style nested representation (Kusupati et al., 2022), concentrating the most important semantics in the early prefix and potentially exacerbating the spectral concentration issue discussed earlier.

We therefore introduce an explicit budget announcement before each <MEM> readout, formatted as <NMEM>n</NMEM><MEM><MEM>$\cdots$, where n denotes the memory budget and <NMEM> and </NMEM> are added special tokens. We prepend the corresponding budget marker to each <MEM> block, which allows each memory embedding to condition on the full budget, enabling the tower to redistribute information across the entire memory matrix. As shown in ablation stuties (§5.2), our budget-dependent encoding turns out important for memory geometry and downstream accuracy.

Rank regularization. Section 3.2 shows that rank limits the independent directions available to linear attention. To encourage variation across these directions, we turn the spectral surrogate of §3.3 into a regularizer with two complementary terms. The first rewards spreading variation across memory directions, without rewarding a larger embedding scale. The second discourages embeddings from collapsing toward their mean, which spectral spread alone cannot prevent:

$$\mathcal{L}_{\text{rank}}(E) \;=\; -\bar{C}(E) \;+\; \left(\frac{\max(0,\; \tau - v)}{\tau}\right)^2, \qquad v \;=\; \frac{\|E - \mathbf{1}\mu\|_F^2}{\|E\|_F^2}. \tag{3}$$

Here $\bar{C} \in [0, 1]$ is the log-determinant surrogate (§3.3) normalized for scale and memory budget; v measures variation around the mean memory embedding μ relative to the matrix's total squared magnitude, and $\tau > 0$ sets its minimum desired level. Appendix A.2 gives the exact spectral normalization. We add a small $\lambda\mathcal{L}_{\text{rank}}$ to the training loss and select λ on development data. The original cross-entropy loss learns context utilization, while the regularizer encourages spectral spread.

4.2 STAGE 2: TASK SPECIALIZATION AT A FIXED RATE

Stage 2 specializes the pretrained model on downstream tasks such as multi-passage QA, instruction following and reasoning. We adopt the distillation objective following Louis et al. (2025): a teacher LLM generates a response upon the regular input; our decoder learns to reproduce this response by cross-entropy loss but with the corresponding memery matrices in the decoder context.

At this stage we fix the encoding rate T/n intended for deployment, focusing adaptation on the tasks and budget at which the model will be used during inference. Budget conditioning remains enabled, and rank regularization is optional. Throughout both stages, the memory tower and connector are trained in full, while we adopt lora-finetuning (Hu et al., 2022) for the decoder to reduce overfit risks. More training details are further provided in Appendix A.2.

5 Main Experiments

We evaluate our model, noted as **MemTower**, in two complementary settings: Wikipedia-based multi-passage QA for comparison with prior works; math and knowledge reasoning for more insightful analysis. We then ablate design choices to examine their contributions to task performance.

LLM backbones. We adopt the Qwen3 family (Yang et al., 2025) for its diverse range of model sizes offered. Our main experiments use models ranging from 1.7B to 8B for both the memory tower and decoder, while we use Qwen3-32B as the teacher to generate distillation targets.

Training data. Our stage 1 adopts the 10B-token version of the SmolLM2 corpus for pretraining (Ben Allal et al., 2025). For stage 2, we prepare two separate training sets. The first follows COCOM and PISCO (Rau et al., 2025; Louis et al., 2025): near 500k questions over Wikipedia-KILT passages of 128 tokens, with teacher answers conditioned on the top five retrieved passages. The second combines general instruction following data from Tulu 3 (Lambert et al., 2025), math reasoning from NuminaMath-1.5 (Li et al., 2024) and PRM800K (Lightman et al., 2024), and knowledge reasoning from MMLU (Hendrycks et al., 2021) and CommonsenseQA (CSQA; Talmor et al., 2019). Details on the two training data configurations are described in Appendix A.4.

Evaluation protocol. Our models are first evaluated on five multi-passage QA datasets including: ASQA (Stelmakh et al., 2022), HotpotQA (Yang et al., 2018), Natural Questions (NQ; Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023), following the same retrieval pipeline and answer-matching metric as Rau et al. (2025); Louis et al. (2025). They serve as in-domain evaluation for models trained on Wiki-based QA. We also extend evaluation beyond Wiki-based QA with a more reasoning-oriented setting to mitigate data contamination, comprising MATH500 (Lightman et al., 2024), GSM8K (Cobbe et al., 2021), CSQA, and MMLU (their test split). Appendix A.5 further details both protocols.

Baselines. For multi-passage QA, we compare with prior works of a similar embedding-based interface, including AutoCompressor, ICAE, xRAG, COCOM, and PISCO (Chevalier et al., 2023; Ge et al., 2024; Cheng et al., 2024; Rau et al., 2025; Louis et al., 2025). We exclude works of non-embedding representation such as MemoRAG (Qian et al., 2025) and Beacon (Zhang et al., 2025), or with query-dependent compression such as COMI (Tang et al., 2026) and OSCAR (Louis et al., 2026), since they are not directly comparable to our plugable memory interface.

5.1 Main results

On the five QA benchmarks (Table 1), MemTower achieves the highest average score among the evaluated memory-based methods at $\times 16$ (0.69), slightly above PISCO (0.68). Both methods approach the full-context references (0.71), indicating that performance in this setting is largely saturated. The train–evaluation overlap documented in Appendix A.6 also limits how directly these scores reflect the models' ability to use passage information through memory. We next evaluate models specialized on the reasoning mixture, encoding each complete problem into memory (Table 2). These tasks probe more than recovery of answer-bearing evidence: solving a problem can depend on precise quantities, relations, and constraints throughout its statement. This dependence on instance-specific details could reduce the effect of memorizing similar QA patterns in training. As such evaluation is not covered by prior related works, we additionally train the best baseline setup PISCO with its released code for comparison in this setting.

At $\times 4$ encoding rate, MemTower with both an 8B tower and decoder surpasses our re-implementation of PISCO on all four benchmarks, averaging 0.79 versus 0.72. The gap to the 8B full-context reference (0.85) widens more drastically at $\times 16$, where accuracy drops to 0.56. These

Table 1: Results for multi-passage QA under the evaluation protocol of Louis et al. (2025). As these Wiki-based data contain similar queries appearing in training and evaluation (Appendix A.6), trained models largely saturate near full-context accuracy, leaving little headroom over the strongest baseline. Table 2 examines more tasks for further insightful comparison of memory utilization.

Method	*Rate*	ASQA	HotpotQA	NQ	TriviaQA	PopQA	Avg.
Full context (Qwen3-8B)	—	0.75	0.54	0.69	0.93	0.64	0.71
Full context (Mistral-7B)	—	0.74	0.51	0.69	0.92	0.70	0.71
AutoCompressor	$\times 4$	0.57	0.31	0.35	0.70	0.24	0.43
ICAE	$\times 4$	0.47	0.29	0.42	0.78	0.43	0.48
xRAG (Mistral-7B)	$\times 128$	0.34	0.27	0.32	0.77	0.33	0.40
COCOM	$\times 16$	0.63	0.46	0.58	0.89	0.48	0.61
PISCO (Mistral-7B)	$\times 16$	**0.72**	0.48	0.65	0.90	**0.66**	0.68
MemTower (Qwen3-8B)	$\times 16$	**0.72**	**0.53**	**0.66**	**0.92**	0.62	**0.69**

Table 2: Accuracy on four reasoning-oriented datasets to mitigate data contamination. Each problem is encoded into a memory matrix in full, and evaluated models are finetuned on the second training set. These benchmarks are not supported by prior models in Table 1; we therefore re-implement and train PISCO with its released code for comparison. Size pairs denote: memory tower $\rightarrow$ decoder.

Configuration	*Rate*	MATH500	GSM8K	CSQA	MMLU	Avg.
Full-context references						
Qwen3-1.7B	—	0.73	0.82	0.70	0.63	0.72
Qwen3-8B	—	0.84	0.94	0.80	0.81	0.85
PISCO: our re-implementation						
8B $\rightarrow$ 8B	$\times 4$	0.72	0.79	0.67	0.67	0.72
MemTower: encoding-rate comparison						
8B $\rightarrow$ 8B	$\times 4$	**0.76**	**0.90**	**0.77**	**0.73**	**0.79**
8B $\rightarrow$ 8B	$\times 16$	0.57	0.62	0.51	0.54	0.56
MemTower: size comparison at a fixed rate						
1.7B $\rightarrow$ 8B	$\times 4$	0.72	0.86	0.75	0.71	0.76
4B $\rightarrow$ 8B	$\times 4$	0.75	0.88	**0.77**	**0.73**	0.78
8B $\rightarrow$ 1.7B	$\times 4$	0.64	0.76	0.68	0.59	0.67

larger gaps suggest a more demanding test of memory retention than multi-passage QA. Meanwhile, a 4B tower with an 8B decoder nearly matches the 8B–8B configuration (0.78 versus 0.79), while reducing decoder size from 8B to 1.7B incurs larger degradation in these comparisons.

5.2 Ablation study

We evaluate ablations on the four reasoning benchmarks in Table 2 to avoid confounding the comparison with the Wikipedia QA overlap discussed in Appendix A.6. Table 3 shows that all three components contribute at both encoding rates. Pretraining improves accuracy on every benchmark, supporting its role in learning to communicate through memory before task specialization. Budget announcement is particularly important for distributing information across memory embeddings at larger memory budgets: removing it lowers average accuracy by 7.9% at $\times 4$ and 4.9% at $\times 16$, relative to the reference at each rate. Conversely, rank regularization becomes more valuable as the budget tightens: removing it yields a relative accuracy drop of 1.6% at $\times 4$, rising to 7.6% at $\times 16$. This pattern is consistent with the role of rank in §3.2: when fewer embeddings represent the same input, redundancy can be more costly, corroborating broader use of the memory matrix's directions.

Autoencoding supervision in pretraining. We also experiment Autoencoding (AE) as a pretraining objective: given the memory matrix produced by the tower, the decoder learns to reconstruct the

Table 3: Ablations of MemTower with the 8B memory tower and decoder, using the four benchmarks in Table 2. Each variant removes one component from the reference method. Avg. reports reference accuracy and, for ablations, the relative percentage decrease at the same rate.

Configuration	*Rate*	MATH500	GSM8K	CSQA	MMLU	Avg.
Reference method	$\times 4$	0.76	0.90	0.77	0.73	**0.79**
w/o pretraining	$\times 4$	0.74	0.88	0.75	0.69	-3.2%
w/o budget announcement	$\times 4$	0.73	0.82	0.69	0.67	-7.9%
w/o rank regularization	$\times 4$	0.76	0.88	0.76	0.71	-1.6%
Reference method	$\times 16$	0.57	0.62	0.51	0.54	**0.56**
w/o pretraining	$\times 16$	0.53	0.57	0.48	0.51	-6.7%
w/o budget announcement	$\times 16$	0.54	0.59	0.48	0.52	-4.9%
w/o rank regularization	$\times 16$	0.53	0.56	0.49	0.49	-7.6%

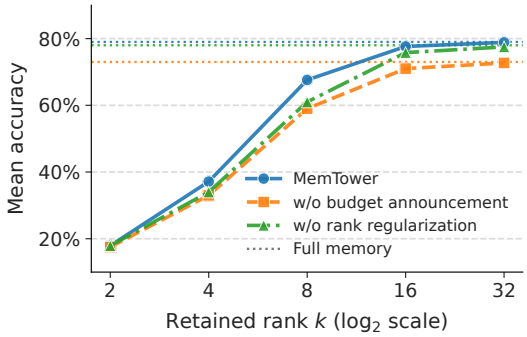


Figure 2: Reasoning accuracy of 8B MemTower at $\times 4$ when retraining k memory directions.

Table 4: Mean task scores at $\times 4$, including our PISCO reimplementation. Usable rank preserves at least 98% of each model's full-memory by average scores.

	Wiki QA	Reasoning	
Rank k	MemTower	MemTower	PISCO
2	0.34	0.18	0.18
4	0.43	0.37	0.38
8	0.54	0.68	0.65
16	0.66	0.78	0.72
32	0.69	0.79	0.72
Full memory	0.69	0.79	0.72
$R^\star$	**32**	**16**	**16**

original source text. We add AE to pretraining, using cross-entropy on the source text as auxiliary supervision. Prior work reports mixed evidence on the downstream value of AE: reconstruction supervision can be beneficial (Ge et al., 2024), yet accurate source recovery does not consistently translate into better task performance (Louis et al., 2025). Our evaluation on four reasoning benchmarks suggests that it depends on the encoding rate: AE provides little benefit at low rates but appears helpful at higher rates (Appendix A.8, Table 9), suggesting that explicit reconstruction objective can help shape the representation in latent space as the memory budget tightens.

6 Rank Analysis

The task ablations establish the benefits of budget announcement and rank regularization. We next ask two complementary questions: whether these components help the decoder extract useful information from more of the memory spectrum, and whether regularization changes the geometry of the representations produced during training.

6.1 Task accuracy and usable rank

We apply the intervention of §3.3 to the 8B MemTower at $\times 4$, retaining $k \in \{2, 4, 8, 16, 32\}$ principal directions per memory matrix while fixing its length, mean, and decoder. This isolates how predictive performance changes as progressively more of an already trained memory representation is restored. Comparison with the component ablations reveals whether their accuracy gaps arise within the dominant directions or from information distributed farther across the spectrum. Table 4 additionally reports our PISCO reimplementation and the separately trained multi-passage QA model; Appendix A.7 provides the protocol.

Table 5: Spectral spread recorded during training (§6.2). Rank regularization broadens the spectrum of both the tower's readout states H and the decoder-facing memory matrix E (Appendix A.9).

Representation	w/o rank regularization	MemTower
Tower hidden states H (before the connector)	0.83	**0.90**
Memory matrix E (after the connector)	0.80	**0.95**

The curves nearly coincide at $k = 2$ and separate as more directions are restored (Figure 2), indicating that the gains of the full method are not confined to the leading directions. The largest gaps occur at $k = 8$: MemTower reaches 0.68 accuracy, compared with 0.59 without budget announcement and 0.61 without rank regularization. Budget announcement produces the larger gain, while regularization becomes especially valuable once the intervention retains enough directions to expose differences beyond the dominant subspace. With a 2% relative tolerance, MemTower has usable rank $R^\star = 16$, whereas both ablations require 32 despite their lower full-memory accuracies. The complete training design therefore reaches a higher accuracy ceiling and recovers that ceiling from fewer retained directions.

MemTower achieves higher reasoning accuracy than PISCO with the same usable rank (Table 4). Both need 16 directions to remain within 2% of their respective full-memory scores, but those directions sustain a higher accuracy for MemTower (0.79 versus 0.72). Usable rank must therefore be interpreted together with the performance it preserves: the same threshold can describe representations with different predictive value.

6.2 How regularization shapes memory geometry

The rank regularizer is designed to prevent memory variation from concentrating in a few dominant directions. We test whether it has this intended geometric effect using measurements recorded in the training logs, on both the tower readout states and the decoder-facing memory matrix. Our measure, *spectral spread*, is a normalized participation ratio that increases as variation is distributed more evenly across the available directions; Appendix A.9 gives its formal definition.

Rank regularization increases spread from 0.80 to 0.95 in the memory matrix after the connector and from 0.83 to 0.90 in the tower states before it (Table 5). Its effect therefore propagates through the connector and reshapes what the tower writes, rather than appearing only in the final projection into the decoder embedding space. These increases reflect a more even distribution of variation across directions rather than a uniform growth in embedding magnitude.

We particularly note that a broader spectrum is necessary for exposing more independent directions to attention, but it is not sufficient for better prediction: additional variation may encode little task-relevant information or remain difficult for the decoder to use. The two analyses therefore play distinct roles. Spectral spread verifies that the regularizer changes the intended geometry, while the accuracy–rank curves determine whether the resulting directions support prediction.

7 Conclusion

We study pluggable memory through both the geometry of its memory matrix and the predictions that geometry supports. A linear attention bound identifies rank as a constraint on the directions available to the decoder, while usable rank measures how many principal directions are needed to preserve task performance. Guided by this view, MemTower combines variable-rate continuation pretraining, budget announcement, and rank regularization with fixed-rate task specialization. Across QA and reasoning tasks, it improves over embedding-based memory baselines; ablations and rank interventions further show that its training design sustains higher accuracy through better use of the available directions. The accompanying increase in spectral spread confirms the intended geometric effect, while also underscoring that spread becomes meaningful only when paired with task behavior. Together, accuracy and usable rank provide a more informative account of learned memory than embedding count alone.

AI USE STATEMENT

In this work, we used OpenAI GPT models to assist with drafting and revising portions of the manuscript, including improving its clarity and presentation. We also used OpenAI Codex to assist with writing selected code used in the experimental workflow and the proof derivation of the proposition. All AI-generated text was reviewed, revised, and verified by the authors. AI-assisted code was inspected and tested for correctness, and the authors independently verified the reported experimental results, mathematical statements, citations, and conclusions. We take full responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the aid of generative AI.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=EbMuimAbPbs.

Md Rifat Arefin, Gopeshh Subbaraj, Nicolas Gontier, Yann LeCun, Irina Rish, Ravid Shwartz-Ziv, and Christopher Pal. Seq-VCR: Preventing collapse in intermediate transformer representations for enhanced reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=30oIfmrcFO.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xm6YD62D1Ub.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martin Blazquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Agustín Piqueres Lajarín, Hynek Kydlíček, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan Son Nguyen, Ben Burtenshaw, Clémentine Fourrier, Haojun Zhao, Hugo Larcher, Mathieu Morlon, Cyril Zakka, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM2: When smol goes big — data-centric training of a fully open small language model. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=3JiCl2A14H.

Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. Matryoshka multimodal models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Uhj5OxAz7I.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision (ECCV)*, 2024. doi: 10.1007/978-3-031-73004-7_2. arXiv:2403.06764.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. xRAG: Extreme context compression for retrieval-augmented generation with one token. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=6pTlXqrO0p.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3829–3846, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.232. URL https://aclanthology.org/2023.emnlp-main.232/.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John

Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 55–65, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1006. URL https://aclanthology.org/D19-1006/.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkEYojRqtm.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann Lecun. RankMe: Assessing the downstream performance of pretrained self-supervised representations by their rank. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10929–10974. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/garrido23a.html.

Tao Ge, Hu Jing, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=uREj4ZuGJE.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.825. URL https://aclanthology.org/2023.emnlp-main.825/.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 30233–30249. Curran Associates, Inc., 2022. doi: 10.52202/068431-2192. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/c32319f4868da7613d78af9993100e42-Paper-Conference.pdf.

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

Association for Computational Linguistics, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, 2023. doi: 10.1145/3600006.3613165. arXiv:2309.06180.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=i1uGbfHHpH.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243/.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. NuminaMath. Hugging Face dataset, 2024. URL https://huggingface.co/datasets/AI-MO/NuminaMath-1.5.

Zongqian Li, Yixuan Su, and Nigel Collier. 500xCompressor: Generalized prompt compression for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25081–25091, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1219. URL https://aclanthology.org/2025.acl-long.1219/.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Guojie Liu, Yiqi Wang, Yanfeng Yang, Wenqi Fan, Songlei Jian, Jianfeng Zhang, and Jie Yu. Cognitive chunking for soft prompts: Accelerating compressor learning via block-wise causal masking. *arXiv preprint arXiv:2602.13980*, 2026. URL https://arxiv.org/abs/2602.13980.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916. Curran Associates, Inc., 2023. doi: 10.52202/075280-1516. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. doi: 10.1109/cvpr52733.2024.02484. arXiv:2310.03744.

Maxime Louis, Hervé Déjean, and Stéphane Clinchant. PISCO: Pretty simple compression for retrieval-augmented generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 15506–15521, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.800. URL https://aclanthology.org/2025.findings-acl.800/.

Maxime Louis, Thibault Formal, Hervé Déjean, and Stéphane Clinchant. OSCAR: Online soft compression for RAG. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ideKAUWvFE.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=2DtxPCL3T5.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, pp. 2366–2377, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400712746. doi: 10.1145/3696410.3714805. URL https://doi.org/10.1145/3696410.3714805.

David Rau, Shuai Wang, Hervé Déjean, Stéphane Clinchant, and Jaap Kamps. Context embeddings for efficient answer generation in retrieval-augmented generation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, WSDM '25, pp. 493–502, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400713293. doi: 10.1145/3701551.3703527. URL https://doi.org/10.1145/3701551.3703527.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European Signal Processing Conference*, pp. 606–610, 2007.

Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *J. ACM*, 54(4):21–es, July 2007. ISSN 0004-5411. doi: 10.1145/1255443.1255449. URL https://doi.org/10.1145/1255443.1255449.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. ASQA: Factoid questions meet long-form answers. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8273–8288, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.566. URL https://aclanthology.org/2022.emnlp-main.566/.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, 2019. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

Sijun Tan, Xiuyu Li, Shishir G Patil, Ziyang Wu, Tianjun Zhang, Kurt Keutzer, Joseph E. Gonzalez, and Raluca Ada Popa. LLoCO: Learning long contexts offline. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17605–17621, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.975. URL https://aclanthology.org/2024.emnlp-main.975/.

Jiwei Tang, Shilei Liu, Zhicheng Zhang, Yujin Yuan, Libin Zheng, Wenbo Su, and Bo Zheng. COMI: Coarse-to-fine context compression via marginal information gain. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=OGDIXDfaN4.

William Timkey and Marten van Schijndel. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In Marie-Francine Moens, Xuanjing Huang,

Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4527–4546, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.372. URL https://aclanthology.org/2021.emnlp-main.372/.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024. URL https://arxiv.org/abs/2409.12191.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=mlJLVigNHp.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Howard Yen, Tianyu Gao, and Danqi Chen. Long-context language modeling with parallel context encoding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2588–2610, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.142. URL https://aclanthology.org/2024.acl-long.142/.

Yaodong Yu, Kwan Ho Ryan Chan, Chong You, Chaobing Song, and Yi Ma. Learning diverse and discriminative representations via the principle of maximal coding rate reduction. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9422–9434. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6ad4174eba19ecb5fed17411a34ff5e6-Paper.pdf.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12310–12320. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/zbontar21a.html.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Long context compression with activation beacon. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=1eQT9OzfNQ.

A Supporting Derivations and Experimental Details

A.1 Proofs

We use two spaces to prove Proposition 1:

$$\mathcal{S}(E) = \{E^\top c : c \in \mathbb{R}^n\} \subseteq \mathbb{R}^d, \qquad \mathcal{P}(E) = \{Eu : u \in \mathbb{R}^d\} \subseteq \mathbb{R}^n. \tag{4}$$

The **read subspace** $\mathcal{S}(E)$ is the row space, containing all linear combinations of memory embeddings. The **pattern subspace** $\mathcal{P}(E)$ is the column space, containing the scores obtained by taking each memory embedding's dot product with a shared direction u. Both have dimension r.

Proof of Proposition 1. For part (a), $E^\top \alpha \in \mathcal{S}(E)$ for every α. The set of attainable outputs is therefore contained in the fixed subspace

$$\mathcal{O}(E) = W_V^\top \mathcal{S}(E) \subseteq \mathbb{R}^{d_v},$$

whose dimension is at most $\dim \mathcal{S}(E) = r$ because a linear map cannot increase dimension. For part (b), write $u(q) = W_K W_Q^\top q$. The attainable score vectors lie in

$$\mathcal{L}(E) = \{Eu : u \in \mathrm{Im}(W_K W_Q^\top)\} \subseteq \mathcal{P}(E).$$

Hence $\dim \mathcal{L}(E) \leq \dim \mathcal{P}(E) = r$. Equality need not hold in either part because the projections may impose a smaller subspace. □

The proposition concerns the total rank of the memory matrix. By contrast, the intervention in Eq. equation 2 retains at most k directions after centering; restoring the mean gives $\mathrm{rank}(E^{(k)}) \leq k + 1$. Thus usable rank indexes retained variation around the mean rather than the exact total rank presented to attention.

Corollary 1 (Linearly dependent memory embeddings do not increase linear dimension)**.** *Appending a row in the span of the existing rows leaves the memory rank and read subspace unchanged. The augmented pattern subspace also has dimension r, although it lies in a larger ambient space. Such an embedding therefore adds no independent linear direction; it can nevertheless change the softmax weights and the resulting output.*

The connector's rank constraint. The attention bound applies to each head and query. The connector further constrains the rank of the memory matrix it supplies:

$$E = \Sigma W_2^\top + \mathbf{1} b_2^\top, \qquad \Sigma = \sigma\big(H W_1^\top + \mathbf{1} b_1^\top\big), \tag{5}$$

where W_1, W_2, b_1, b_2 are its parameters and σ is SiLU. The final projection satisfies $\mathrm{rank}(\Sigma W_2^\top) \leq \min(\mathrm{rank}\,\Sigma, \mathrm{rank}\,W_2)$, while the bias term has rank at most one. Together with the matrix dimensions, this gives

$$r \leq \min\{n, d, \min(\mathrm{rank}\,\Sigma, \mathrm{rank}\,W_2) + 1\}.$$

For the centered geometry used by our regularizer and rank intervention, let $P = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$. Since $P\mathbf{1} = 0$, centering removes the connector bias and yields

$$PE = P\Sigma W_2^\top, \qquad \mathrm{rank}(PE) \leq \min\{n-1, d, \mathrm{rank}(P\Sigma), \mathrm{rank}\,W_2\}.$$

Thus both the intermediate representation and the final projection constrain the total and centered rank available at the memory interface.

Rank under row and channel rescaling. Consider the transformation $E \mapsto DE\Gamma$, where $D = \mathrm{diag}(1/\rho_i)$ and $\Gamma = \mathrm{diag}(\gamma)$ are invertible diagonal matrices. Then $\mathrm{rank}(DE\Gamma) = \mathrm{rank}(E)$, and the read and pattern subspaces become $\Gamma\mathcal{S}(E)$ and $D\mathcal{P}(E)$, respectively. This applies to RMS-style normalization when row denominators are finite and nonzero and channel gains are nonzero; a positive numerical stabilizer in the row denominator does not itself alter the factorization. The statement concerns this normalization step only. It does not establish rank preservation through position-dependent key rotations, additive biases, or subsequent nonlinear blocks.

Table 6: Training components in MemTower and their motivation. Stage 2 uses teacher-generated responses for each task mixture.

component	stage 1	stage 2	motivation
objective	continuation	teacher distillation	preserve context useful for downstream prediction
rate sampling	log-uniform, 4–64	fixed rate	learn across budgets, then specialize at deployment rate
budget announcement	on	on	condition the representation on the available memory budget
rank regularizer	on, tuned λ	on (optional)	discourage spectral concentration while retaining predictive content
reconstructive AE	optional	none	provide auxiliary supervision for context retention
memory tower	full fine-tune	full fine-tune	adapt language-model states to memory encoding
decoder	LoRA	LoRA	adapt memory reading with limited parameter updates
connector	trained, zero-init W_2	trained	map memory tower states to decoder embeddings

A.2 Training configuration and parameterization

Parameterization. We initialize W_2 to zero and b_2 to the mean decoder input embedding.

We fully fine-tune the memory tower and train the connector throughout. The decoder uses LoRA (Hu et al., 2022), with input embeddings also trainable to accommodate the added tokens. The <MEM> token and budget delimiters <NMEM> and </NMEM> are initialized from the mean input embedding of their respective models and learned during training.

Spectral normalization in the rank regularizer. The regularizer in equation 3 separates the distribution of variation across directions from its magnitude relative to the memory mean. Let $Z_0 = E - \mathbf{1}\mu$ be the centered memory matrix. For $n > 1$ and nonzero centered variance, define

$$\hat{G} = \frac{nZ_0Z_0^\top}{\|Z_0\|_F^2}, \qquad \bar{C}(E) = \frac{\log\det(I + \hat{G}/\alpha)}{R_{\max}(n,\alpha)}, \qquad R_{\max}(n,\alpha) = (n-1)\log\left(1 + \frac{n}{\alpha(n-1)}\right). \tag{6}$$

The matrix $Z_0Z_0^\top$ contains pairwise inner products of the centered memory embeddings; its eigenvalues are their squared singular values. Normalizing its trace to n removes overall scale, so increasing $\bar{C}$ requires spreading variation across directions rather than increasing all embedding magnitudes. The parameter $\alpha > 0$ sets the spectral scale, as in §3.3.

Centering leaves at most $n - 1$ nonzero directions. With their total eigenvalue fixed at n, concavity of the logarithm bounds the log-determinant by $R_{\max}$, attained when these directions have equal eigenvalues if $d \geq n - 1$. Dividing by this bound puts $\bar{C}$ in $[0, 1]$ across memory budgets. Because this term is insensitive to scale, equation 3 also penalizes a variance fraction $v < \tau$: the memory embeddings must retain differences relative to their overall magnitude, not merely distribute arbitrarily small differences evenly.

A.3 Hyperparameters

The variable-rate sampler draws between encoding rates 4 and 64 on a logarithmic scale. Its schedule mixes in the largest allowed memory budget to facilitate easy-to-hard curriculum learning, with a probability of largest memory budget that decays during training from 100% to 10%. For the rank regularizer, we set $\alpha = 1$ in equation 6, and we pretrained models with three λ configurations by 0.01, 0.05 and 0.1 for the auxiliary loss in equation 3. Experiments on the development data suggests 0.01 is the best setting with least disturbance to the cross-entropy loss, while still able to encourage spectrum spread.

Table 7: Composition of the instruction/reasoning training set. Sample counts are specified separately for each source.

Category	Source	Examples
General instruction following	Tulu 3 (Lambert et al., 2025)	450k
Mathematics	NuminaMath-1.5 (Li et al., 2024)	50k
	PRM800K, training split (Lightman et al., 2024)	10k
Knowledge and reasoning	MMLU, training split (Hendrycks et al., 2021)	50k
	CSQA (Talmor et al., 2019)	≈20k

Our training is conducted on Nvidia A100 GPUs with a total batch size of 320. We adopt learning rate of 2×10^{-4} for the lora decoder, 5×10^{-4} for the connector, and 2×10^{-5} for the memory tower, with a cosine learning rate schedule. The training runs for one epoch for the stage 1 pretraining; for stage 2 finetuning, we set the max training steps to be 6k.

A.4 Training data

Stage 1 uses the 10B-token version of the SmolLM2 pretraining corpus (Ben Allal et al., 2025). Stage 2 specializes separate models on the two training sets below, using the teacher-distillation objective in §4.2. The teacher is Qwen3-32B throughout.

Wikipedia QA. We adopt the 453k-question training set used by COCOM and PISCO (Rau et al., 2025; Louis et al., 2025). The released question set is `dmrau/multi_qa`[1]; the retrieval collection contains Wikipedia-KILT documents divided into 128-token passages.[2] For each question, the five highest-ranked passages are supplied to the teacher together with the question to generate the distillation target (the retrieval pipeline for training is identical to the one described in evaluation protocols; see Appendix A.5). The memory tower encodes the passages independently, while the decoder receives their memory embeddings of these passages and the plain text question.

Instruction and reasoning. Table 7 lists the sampled sources. This mixture broadens task specialization from retrieving facts in Wikipedia passages to following instructions and solving mathematical and knowledge-based problems. The full problem is encoded into a memory matrix, and the decoder is similarly trained on the teacher response to its original problem.

A.5 Task evaluation protocols

In-domain multi-passage QA. We follow the five-dataset protocol of Louis et al. (2025): ASQA (Stelmakh et al., 2022), HotpotQA (Yang et al., 2018), Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023). SPLADE-v3 retrieves from the 128-token Wikipedia-KILT collection, and a DeBERTa-v3 reranker selects the top five passages, ordered by relevance. The teacher and full-context reference receive the original passages; memory-based methods replace them with their memory representations. The question is supplied as text. Following the prior protocol, decoding is greedy with a maximum of 128 generated tokens.

The answer-match metric follows the same scoring as (Louis et al., 2025), which assigns a score of one when a normalized reference answer occurs within the normalized prediction, taking the maximum over accepted reference answers. Normalization lowercases text, removes punctuation and the articles "a," "an," and "the," and standardizes whitespace. We average over examples within each dataset and report the unweighted mean of the five dataset scores.

Mathematics and knowledge reasoning. We evaluate the model specialized on the instruction/reasoning mixture on MATH500 (Lightman et al., 2024), GSM8K (Cobbe et al., 2021), CSQA (Talmor et al., 2019), and MMLU (Hendrycks et al., 2021). The memory tower encodes the complete

[1] `https://huggingface.co/datasets/dmrau/multi_qa`
[2] `https://huggingface.co/datasets/dmrau/kilt-128`

Table 8: Query overlap between the Wikipedia QA training set and the five evaluation sets. Lexical overlap follows the normalized matching rule in Appendix A.6; semantic overlap applies the additional embedding-similarity threshold. Values are percentages of evaluation questions.

Evaluation set	Split	n	Lexical (%)	Semantic (%)
ASQA	dev	948	100.00	100.00
HotpotQA	dev	5,600	0.68	0.64
NQ	dev	2,837	26.93	20.94
TriviaQA	dev	5,355	33.93	31.91
PopQA	test	14,267	3.79	1.50

problem, including answer options for multiple-choice tasks; no retrieval is used. The full-context teacher reference receives the same problem as textual input.

For mathematics, we score the final answer with the standard `math-verify` package; for CSQA and MMLU, we score the selected answer option. We report each benchmark separately and their unweighted mean. Encoding rates are fixed during evaluation, with the same input content and decoding settings used across methods at each comparison point.

Baseline configurations. AutoCompressor and ICAE use rate 4, xRAG uses rate 128, and COCOM and PISCO use rate 16. The xRAG and PISCO rows use Mistral-7B; PISCO is specifically the Mistral model at $\times 16$. These published results retain their original training and backbone configurations, rather than being retrained on Qwen3. AutoCompressor and xRAG are limited to one context in the source comparison.

A.6 Train–evaluation overlap in Wikipedia QA

We compare the questions in the Wikipedia QA specialization set, `dmrau/multi_qa`, against the five evaluation sets. We first lowercase each question, retain only alphanumeric characters, and collapse whitespace. A pair is counted as a lexical match if the normalized strings are identical, or if both token Jaccard similarity and character-level similarity exceed 0.7. A semantic match additionally requires cosine similarity of at least 0.95 between BGE-BASE-EN-V1.5 embeddings. We search the full training pool rather than sample candidates, using a rare-token inverted index for lexical matching and exact Faiss search for semantic matching.

As Table 8 shows, overlap is substantial for ASQA, NQ, and TriviaQA, but limited for HotpotQA and PopQA. The largest rates reflect known source relationships: ASQA questions originate from NQ-Open training data included by the specialization pool, which also contains NQ-Open and FreebaseQA questions related to the NQ and TriviaQA evaluations. Together with the decoder's internal knowledge, this overlap makes near-full-context QA performance difficult to attribute solely to information conveyed through the memory matrices. We therefore use these results chiefly for direct comparison under the established multi-passage QA protocol and complement them with the mathematics and knowledge-reasoning evaluation in the main text.

A.7 Rank-analysis protocol

Models and inputs. All rank sweeps use an 8B tower and decoder at $\times 4$. The reasoning comparison includes MemTower, its two component ablations, and our PISCO reimplementation from Table 2; each encodes the complete problem, including answer choices where applicable. The separate Wikipedia-QA model encodes the retrieved passages and receives the question as text. Its full-memory mean is approximately 0.69, matching the $\times 16$ QA result in Table 1, although the rank sweep uses $\times 4$.

Intervention and aggregation. We apply equation 2 independently to every memory matrix at $k \in \{2, 4, 8, 16, 32\}$, preserving the matrices' lengths and order. The checkpoint, evaluation examples, memory budgets, and decoding settings remain fixed within each sweep. The truncated centered matrix has rank at most k; restoring the mean can increase total rank to $k + 1$. At $k = 0$,

Table 9: Effect of AE supervision on the four reasoning benchmarks in Table 2, using a 8B memory tower and decoder. Reference rows show MemTower's continuation-only pretraining results; +AE adds the reconstruction objective during pretraining.

Pretraining	*Rate*	MATH500	GSM8K	CSQA	MMLU	Avg.
Continuation	$\times 4$	0.76	0.90	0.77	0.73	0.79
+ AE	$\times 4$	0.76	0.91	0.76	0.72	0.79
Continuation	$\times 16$	0.57	0.62	0.51	0.54	0.56
+ AE	$\times 16$	0.59	0.65	0.53	0.54	0.58

every embedding equals the matrix mean. If $k \geq \min(n-1, d)$, that matrix is recovered in full; rank 32 need not recover every matrix. Scores are unweighted means over the five QA or four reasoning benchmarks, following Appendix A.5.

Usable-rank threshold. We report the smallest tested k satisfying $a(k) \geq 0.98 a_{\text{full}}$, without interpolation. Here *full memory* denotes the original memory matrices, whereas the full-context baseline supplies the original source text. The threshold is applied to each model's averaged mean accuracy, not to individual benchmark thresholds.

A.8 Autoencoding across encoding rates

Table 9 compares continuation pretraining with and without additional AE supervision under the training and evaluation setting of Table 2, as discussed in §5.2. The AE variant samples source-reconstruction examples with probability 0.3 which decays to 0.05 near the end of training, applying next-token cross-entropy to the original source span. In practice, the AE objective quickly converges, suggesting that it is an easier task than the continuation task. As shown by Table 9, adding AE appears more helpful for high encoding rate, while offering trivial boost for low rate scenarios.

A.9 Measuring memory geometry

Table 5 summarizes diagnostics from the training logs of runs with and without rank regularization at $\times 16$. The same diagnostic is applied to the tower readout states H and the decoder-facing memory matrix E. For $X \in \{H, E\}$, let λ_i be the squared singular values after subtracting the mean embedding. Spectral spread is

$$\text{Spread}(X) = \frac{1}{n-1} \frac{\left(\sum_i \lambda_i\right)^2}{\sum_i \lambda_i^2}. \tag{7}$$

The participation ratio measures effective spectral dimensionality. For nonzero variation and embedding width at least $n-1$, normalized spread ranges from $1/(n-1)$ when variation lies in one direction to 1 when the $n-1$ squared singular values are equal. The diagnostic reports zero for identical embeddings and excludes matrices with $n < 2$.

Spread is computed per matrix and averaged in the training logs. It is scale-invariant and complements the normalized log-determinant surrogate $\bar{C}$ used for training (Appendix A.2); both favor a more even spectrum, while neither directly measures task utility.