MANTA: INFORMATION-ADAPTIVE CONTEXT COMPRESSION VIA MATRYOSHKA TOKENS AND TOLERANCE-CONDITIONED ALLOCATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Context compression enables large language models to process long inputs more efficiently by encoding the original context into compact, reusable representations. However, most existing methods fix a single compression ratio at training time, while a few support several predefined ratios. Either way, every context of a given length receives the same allocation, regardless of its information content, resulting in inefficient resource utilization. To address this, we introduce *information-adaptive context compression*: the allocation grows for information-dense contexts and shrinks for redundant ones. We formalize this through a criterion we call the *minimal sufficient allocation*: the smallest allocation whose excess loss, measured against the full allocation, remains below a global tolerance. We realize this criterion in MANTA, which exposes the compression ratio as a continuous inference-time condition, built from two components: (1) Matryoshka tokens, a compact nested representation of each chunk with valid arbitrary-length prefixes; (2) a lightweight Elastic Allocator, conditioned on the tolerance, that decides how many Matryoshka tokens each chunk retains from the excess loss its content would incur. MANTA consistently outperforms existing context compression methods across different compression ratios, improving average token-F1 by 9.3% relative in domain and 5.4% out of domain over the strongest compression-matched baseline. Further analyses attribute the improvements to information-adaptive compression, where the allocator flexibly distributes Matryoshka tokens based on context content.

1 INTRODUCTION

Large language models increasingly leverage rich contextual information (Brown et al., 2020; Grattafiori et al., 2024), and are expected to operate over long contexts such as documents retrieved to ground answers in external knowledge (Lewis et al., 2020; Karpukhin et al., 2020), accumulated dialogue histories (Xu et al., 2022; Maharana et al., 2024), and entire codebases (Zhang et al., 2023a; Liu et al., 2024b). Such length dominates inference cost, where attention and the key–value (KV) cache grow with the number of tokens, and strains a model's use of mid-context evidence (Liu et al., 2024a). Context compression addresses this by re-encoding a long context into a compact compressed representation that replaces the original tokens inside the model. The representation is computed once and reused across queries over the same context, cutting computation and memory while preserving strong downstream performance (Mu et al., 2023; Chevalier et al., 2023; Ge et al., 2024; Li et al., 2025; Liu et al., 2026).

The governing question in context compression is how small a compressed representation can be while still equipping the model with the ability to answer correctly. In practical scenarios, the answer largely varies with the information content of the text: a context introducing several new entities needs a far larger allocation than one carrying little new information, as is shown in Fig. 1a. However, existing context compressors largely ignore this heterogeneity, committing to one allocation at training time and applying it uniformly whatever a context contains (Ge et al., 2024; Li et al., 2025; Zhao et al., 2025; Liu et al., 2026). This has two drawbacks: (1) a uniform compression ratio over-provisions redundant contexts and starves dense ones, sacrificing accuracy at any average compression ratio, (2) the ratio is baked into the weights in advance, so every new ratio means training

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.

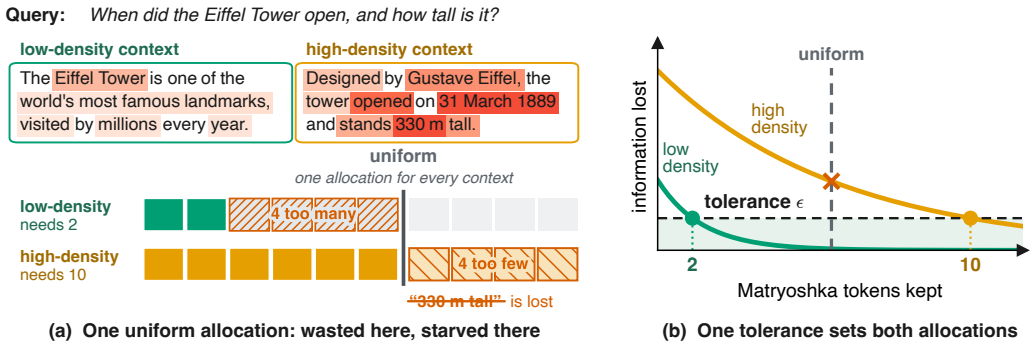

Figure 1: Uniform vs. information-adaptive context compression. (a) A uniform allocation over-provisions sparse contexts and starves dense ones, dropping a queried fact; MANTA gives each what its content requires. (b) One tolerance, met at each context's own minimal sufficient allocation.

and storing another model. Training with several compression ratios lets one model serve multiple predefined allocations (Zhang et al., 2025), but the served ratio is still specified externally rather than derived from the content.

To handle the problem, we therefore introduce *information-adaptive context compression*: what a context keeps is determined by its own content, and its compression level is continuously controlled at inference time rather than fixed during training. To make this precise, we propose the criterion of *minimal sufficient allocation*: each context keeps the smallest allocation whose excess loss, against its own full allocation, stays below one global tolerance. The tolerance is a single distortion constraint shared by all contexts, so one scalar induces different compression ratios for distinct contents, and the average ratio emerges from the context instead of being imposed on it (Fig 1b). We realize the principle by proposing MANTA, **MA**tryoshka **N**esting with **T**olerance-conditioned **A**llocation, which is built from two components. (1) First, we carry prefix nesting (Kusupati et al., 2022) to construct compressor as the *Matryoshka tokens* in place of a chunk, in which arbitrary length of prefix is a valid compressed representation, so one model spans all compression ratios. (2) The Elastic Allocator, a lightweight head conditioned on the tolerance, then decides the allocation of each context based on the excess loss, moving along the nested representation to vary the compression ratio without retraining.

We conduct comprehensive experiments to show that a single MANTA model outperforms dedicated per-ratio baselines across wide range of compression ratios, both in and out of domain, improving average token-F1 by 9.3% relative in domain and 5.4% out of domain over the strongest baseline; the gains extend to contexts far longer than those seen during training and hold across different model scales and backbone families. Analyses further attribute them to adaptive allocation corresponding to information in the content. Our contributions are summarized as follows:

- In contrast to existing context compression methods that fix uniform allocation during training, we formulate *information-adaptive context compression* that allocation follows each context's content, and compression ratio becomes a continuous inference-time condition.
- We introduce *minimal sufficient allocation* and propose MANTA: Matryoshka tokens, arbitrary length of prefix is a valid compressed representation, and a tolerance-conditioned Elastic Allocator that reads a chunk's allocation off its content.
- We perform extensive experiments to show that a single MANTA model consistently outperforms per-ratio baselines at different compression ratios, across in and out of domain, longer contexts, other scales and backbones, showing effectiveness of adaptive allocation.

2 Related Work

Context Compression. A long context can be re-encoded into a compact representation that replaces its tokens for the decoder. The dominant recipe appends dedicated compression tokens that

absorb an arbitrary context (Wingate et al., 2022; Mu et al., 2023; Chevalier et al., 2023; Zhao et al., 2025), including ICAE (Ge et al., 2024) and 500xCompressor (Li et al., 2025); SAC (Liu et al., 2026) instead marks tokens of the context itself with a learned embedding and aggregates each chunk into their key–value states. Encoder-side systems keep the compressed context outside the decoder's cache (Yen et al., 2024; Cheng et al., 2024), while hard-prompt methods delete tokens a small model deems uninformative (Li et al., 2023; Jiang et al., 2023; 2024; Pan et al., 2024), corroborating that compressibility varies across a context but capping compression at whole-token granularity. Parts of the fixed-ratio commitment have been relaxed, though none into query-independent, content-driven adaptivity: Activation Beacon (Zhang et al., 2025) samples a compression ratio per chunk during training so one model serves several, but the served ratio stays externally specified, and its trained grid tops out at $32\times$, well below the ratios studied here, which rules it out as a baseline; COMI (Tang et al., 2026) does allocate unevenly within a context, yet scores its units against the query, so the resulting compressed representation is specific to that query rather than reusable across queries. A separate, orthogonal line, KV-cache eviction (Zhang et al., 2023b; Li et al., 2024; Xiao et al., 2024), prunes the decoder's cache over the *raw tokens* by decoding-time attention statistics, and applies on top of a compressed cache, so the two families compose rather than compete. Architecturally, all of these tie the size of a chunk's compressed representation to something outside the content it carries: a ratio frozen into the weights during training, a small predefined grid of ratios, or the query at hand. Hence an information-dense chunk gets no more room than a redundant one of the same length, and any ratio beyond the training-time grid costs another trained model. MANTA removes this coupling in the architecture itself. Matryoshka tokens make every prefix of a chunk's compressed representation a valid key–value cache, so all compression ratios coexist within one set of weights, and the tolerance-conditioned Elastic Allocator sets the prefix length per chunk from the content alone, which keeps the compressed representation query-independent and reusable while turning the compression ratio into a continuous inference-time condition.

Nested Representation Learning. Matryoshka Representation Learning (Kusupati et al., 2022) supervises every nested prefix of embedding jointly, so one vector serves a spectrum of cost–accuracy trade-offs; the principle has since reached visual tokens (Cai et al., 2025) and the feature dimension of the KV cache (Lin et al., 2025). In all of these the operating point is set externally; none learns it from the content being represented. MANTA brings nesting to a new scenario, the Matryoshka tokens that carry a compressed context, and is, to our knowledge, the first query-independent context compressor whose per-chunk compression ratio follows the content of the text and is tunable at inference through a continuous tolerance.

3 Method

3.1 Preliminaries and Motivation

Preliminaries. An input of T tokens is split into n fixed-length *chunks* of C tokens. A *context compressor* replaces each chunk with at most m *Matryoshka tokens* (Fig. 2), whose per-layer key–value states enter the decoder cache in place of the original tokens. Keeping k_i tokens compresses chunk i by C/k_i, and we call k_i its *allocation*. Uniform compression fixes the same k for all chunks during training, whereas information-adaptive compression chooses k_i for each chunk at inference. Let $\ell_i(k)$ denote the decoder LM loss when chunk i retains k tokens. We assume that $\ell_i(k)$ decreases with diminishing returns.

Motivation. Uniform allocation would be optimal only if every chunk required the same allocation, and it does not: chunks differ widely in length-normalized surprisal, and erasing a chunk leaves the gold answer's loss essentially unchanged for most while a few carry the evidence, so one uniform k must both over-provision and starve.

3.2 Adaptive Allocation as Tolerance-Conditioned Sufficiency

We define the *relative excess loss* $\delta_i(k)$ of chunk i at allocation k, and the *minimal sufficient allocation* as the smallest k whose relative excess loss remains below a global tolerance ϵ:

$$\delta_i(k) \;=\; \frac{\ell_i(k) - \ell_i^{\mathrm{ref}}}{\ell_i^{\mathrm{ref}}}, \qquad k_i^{\star}(\epsilon) \;=\; \min\big\{\, k \in \{1,\dots,m\} \;:\; \delta_i(k) \;<\; \epsilon \,\big\}. \tag{1}$$

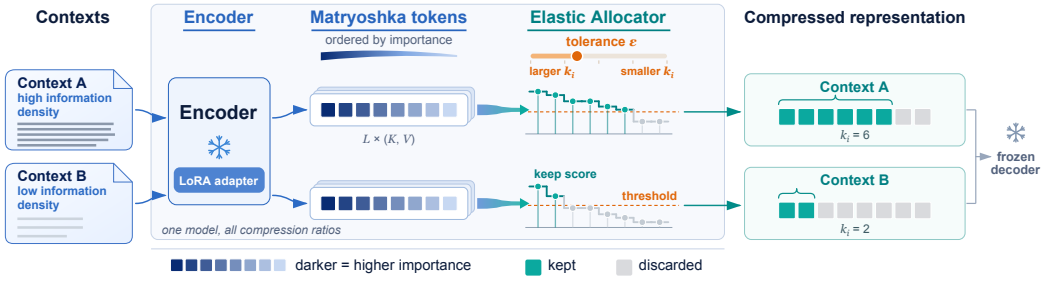

Figure 2: Overview of MANTA. A LoRA-adapted copy of the frozen backbone encodes each context into m importance-ordered Matryoshka tokens, so every nested prefix k is a valid compressed representation at ratio C/k. The Elastic Allocator reads k off the content and the tolerance ϵ.

Here, $\ell_i^{\text{ref}} = \ell_i(m)$ denotes the loss under the full allocation. This makes ϵ a dimensionless fractional tolerance and defines sufficiency relative to the full Matryoshka allocation rather than the uncompressed context. In particular, ϵ bounds the additional loss incurred when truncating from m Matryoshka tokens to k. Notably, the decision is made independently for each chunk, while ϵ is shared globally. As a result, chunks with steeper curves receive larger k, whereas flatter curves receive smaller k. The total allocation $\sum_i k_i$ is not explicitly constrained and instead emerges from the shared ϵ, which is calibrated to a target average ratio.

3.3 Matryoshka Tokens

Matryoshka Token Encoding. Rather than appending dedicated compression tokens, we designate selected context tokens as Matryoshka tokens. We choose m positions spaced one per C/m tokens and mark them with a learned embedding $\boldsymbol{e}_{\text{mt}}$ added to their input embeddings. These tokens are re-encoded by an encoder, which is a LoRA-adapted (Hu et al., 2022) copy of the frozen backbone with bidirectional attention. This allows each token to access the entire chunk. The compressed representation of chunk i consists of their per-layer KV states over L layers of width d:

$$\mathcal{M}_i \;=\; \big\{\,(\boldsymbol{K}_i^{(l)},\,\boldsymbol{V}_i^{(l)}) \in \mathbb{R}^{m\times d}\,\big\}_{l=1}^{L}, \qquad \mathcal{M}_i[{:}k] \;=\; \big\{\,(\boldsymbol{K}_i^{(l)}[{:}k],\,\boldsymbol{V}_i^{(l)}[{:}k])\,\big\}_{l=1}^{L}. \tag{2}$$

The frozen decoder attends to any prefix $\mathcal{M}_i[{:}k]$ using standard causal attention, just as it attends to raw-token KV states. All tokens retain their original absolute position IDs. Changing k_i therefore does not shift the rotary position embedding (RoPE) phase of any other token (Su et al., 2021).

Nesting Objective. To make every prefix a valid compressed representation, we optimize the LM loss jointly over a ladder of nested prefix sizes at each step:

$$\mathcal{L}_{\text{Mat}} \;=\; \textstyle\sum_{k\in K_{\text{set}}} w_k \cdot \mathcal{L}_{\text{LM}}\big(\text{decoder}\big(\{\mathcal{M}_i[{:}k]\}_{i=1}^{n}\big) \to \text{target}\big), \tag{3}$$

Here, K_{set} is the Matryoshka ladder consisting of a fixed set of nested prefix sizes, and w_k denotes endpoint-protective weights, which upweight the ladder's two ends, $k{=}1$ and $k{=}m$, so that the extreme operating points are not underfit. Training each prefix jointly encourages earlier positions to retain more important information. Untrained sizes $k \notin K_{\text{set}}$ also interpolate smoothly, allowing the allocator to output any integer. Additional explanations are shown in App. A.

3.4 Elastic Allocator and Training

Architecture. The allocator maps each chunk's Matryoshka-token KV states together with the tolerance ϵ to an allocation k_i. For each Matryoshka token s, the K and V states from all layers are projected to low-dimensional representations, concatenated, and mapped to a hidden feature. An embedding of ϵ is then added to condition the allocation.

$$\boldsymbol{h}_s \;=\; \boldsymbol{W}_h\big[\textstyle\bigoplus_{l=1}^{L}(\boldsymbol{W}_l^K \boldsymbol{k}_s^{(l)} \oplus \boldsymbol{W}_l^V \boldsymbol{v}_s^{(l)})\big] \;+\; \text{MLP}_\epsilon(\epsilon), \qquad \boldsymbol{a}_{1:m} \;=\; \text{SelfAttn}(\boldsymbol{h}_{1:m}), \tag{4}$$

A single self-attention layer models interactions among the chunk's m Matryoshka tokens. A per-token head then predicts a keep probability, and the cumulative minimum enforces the monotone decisions required by nesting, since a Matryoshka token can be kept only if all earlier ones are:

$$p_s = \sigma(\boldsymbol{w}^\top \boldsymbol{a}_s), \qquad q_s = \min_{t\le s} p_t, \qquad k_i \;=\; \max\big(1,\; \#\{\,s : q_s > 0.5\,\}\big). \tag{5}$$

The allocator is lightweight and requires no additional re-encoding. The decoder directly uses the first k_i Matryoshka tokens from each chunk. Allocation is query-independent by design because the allocator observes only the chunk content and ϵ, which keeps compressed representations reusable.

Training with Dynamic Tolerance. Training labels are provided by a per-chunk oracle that applies the criterion of Eq. 1 over the ladder:

$$k_i^{\text{oracle}}(\epsilon) \;=\; \min\big\{\, k \in K_{\text{set}} \;:\; \hat{\delta}_i(k) \;<\; \epsilon \,\big\}, \qquad \hat{\delta}_i(k) \;=\; \frac{\ell_i(k) - \ell_i^{\text{ref}}}{\max\big(\ell_i^{\text{ref}},\, 10^{-3}\big)}, \tag{6}$$

where $\hat{\delta}_i$ is the relative excess loss δ_i of Eq. 1 with its denominator floored at 10^{-3} for numerical stability, and the losses $\ell_i(k)$ are evaluated without gradients while all other chunks use the allocator's own predictions (App. A.1). Oracle labels are searched over the ladder K_{set}, limiting the cost to $|K_{\text{set}}|$ forwards per chunk. Since the head predicts a keep probability for each Matryoshka token, the resulting k_i in Eq. 5 can take any integer value in $\{1, \ldots, m\}$. We train the head with per-token binary cross-entropy against $y_s = \mathbf{1}[s \le k_i^{\text{oracle}}]$:

$$\mathcal{L}_{\text{alloc}} \;=\; -\tfrac{1}{m} \textstyle\sum_{s=1}^{m} \big[\, y_s \log p_s + (1 - y_s) \log(1 - p_s) \,\big], \qquad \mathcal{L} \;=\; \mathcal{L}_{\text{Mat}} + \alpha\, \mathcal{L}_{\text{alloc}}. \tag{7}$$

We ramp α during training and detach the allocator inputs, so the head sends no gradient to the encoder and the compressed representation is shaped by the ladder objective alone. The key design is **dynamic** ϵ. At each step, we sample ϵ from a grid and use it both to condition the allocator (Eq. 4) and to define the oracle target (Eq. 6). Under a single fixed ϵ, the allocator would see one target per chunk and could ignore its ϵ input; sampling ϵ exposes it to the whole family of oracle allocations, so it learns how k_i moves with the tolerance. This joint conditioning enables effective inference-time control through ϵ. More analysis is shown in §4.4.

Two-Stage Training. We train the compression modules in two stages. Stage 1 pretrains them on a general-domain corpus with the ladder objective in Eq. 3; the allocator is held out at the beginning and ramped in over the later part of the stage, once the nested representation has formed. Stage 2 fine-tunes the same parameters on a question-answering mixture, with the allocator trained throughout. Only the encoder LoRA and the allocator are trainable. The decoder remains frozen, which keeps the compressed representations compatible with raw-token KV states. Further details are provided in App. A.1.

4 Experiments

4.1 Experimental Setup

Dataset and Evaluation Metrics. We pretrain on the long-form corpus PG-19 (Rae et al., 2020), and finetune and evaluate on the MRQA (Fisch et al., 2019) question-answering benchmark, reporting both of its test splits, six in-domain (ID) and six out-of-domain (OOD) datasets. We additionally evaluate on needle-QA at 4K–32K tokens, far beyond the training length. Metrics are SQuAD-normalized (Rajpurkar et al., 2016) token-F1 and exact match (EM), macro-averaged within each split. More details are shown in App. A.2.

Implementation Details. MANTA uses Llama-3.2-1B (Grattafiori et al., 2024) as both encoder and decoder, with Llama-3.2-3B, Llama-3.1-8B and Qwen2.5-1.5B (Yang et al., 2024) in the generalization studies. The encoder's LoRA (Hu et al., 2022) adapters and the Elastic Allocator are trainable, and the decoder stays frozen. Each context is partitioned into chunks of C=512 tokens, each encoded into m=32 importance-ordered Matryoshka tokens of which the allocator retains a prefix of any length $k_i \in \{1, \ldots, 32\}$, trained over the ladder $K_{\text{set}} = \{1, 2, 4, 8, 16, 32\}$, so a single model covers $16\times$ to $512\times$. Both training stages resample the tolerance ϵ at every step. Complete hyperparameters are in App. A.1.

Baselines. We compare against three soft context compression methods, ICAE (Ge et al., 2024), 500xCompressor (Li et al., 2025), and SAC (Liu et al., 2026). Each of which commits to one compression ratio during training and is therefore trained separately for every ratio. Also, we report the backbone finetuned on uncompressed contexts as an uncompressed reference. All baselines are retrained on the same data under the same protocol as MANTA and scored by the identical scorer.

Table 1: Main results at 64× compression on full MRQA. MANTA uses one adaptive model at ϵ=0.05, while baselines are trained per ratio.

(a) MRQA in-domain

Methods	Ratio	SQuAD		NewsQA		TriviaQA		SearchQA		HotpotQA		NQ		Average	
		F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
Full-FT	1×	91.5	85.0	68.3	52.7	82.4	78.1	88.1	83.3	80.9	65.5	79.2	67.5	81.73	72.01
ICAE	64×	34.2	22.4	24.6	13.3	57.9	52.1	69.1	62.5	39.0	25.3	49.5	33.6	45.70	34.85
500xC	64×	38.4	26.4	29.1	17.8	59.2	53.2	76.2	70.4	49.4	34.2	54.0	39.5	51.04	40.25
SAC	64×	37.0	25.5	31.9	19.3	65.1	59.2	**78.7**	**72.4**	47.9	32.1	51.4	37.1	52.01	40.93
MANTA	**65×**	**46.5**	**33.5**	**35.8**	**22.1**	**65.2**	**59.3**	78.1	72.0	**57.5**	**41.0**	**58.0**	**43.1**	**56.87**	**45.17**

(b) MRQA out-of-domain

Methods	Ratio	BioASQ		DROP		DuoRC		RACE		RelExt		TbQA		Average	
		F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
Full-FT	1×	75.4	62.9	53.8	44.8	65.3	55.8	51.5	39.5	86.5	75.6	67.0	58.7	66.59	56.21
ICAE	64×	46.3	35.7	31.1	22.5	16.7	11.6	19.5	12.0	65.4	52.9	26.5	20.0	34.26	25.78
500xC	64×	46.4	38.8	32.3	24.9	16.3	11.0	20.9	11.6	**71.3**	**59.2**	34.9	28.2	37.02	28.94
SAC	64×	54.0	43.2	35.7	26.9	27.6	21.2	23.1	13.2	55.5	42.8	**44.8**	**36.3**	40.10	30.59
MANTA	**65×**	**55.2**	**45.3**	**35.8**	**27.0**	**30.3**	**23.5**	**26.3**	**16.0**	64.5	52.0	41.5	33.7	**42.25**	**32.92**

4.2 Main Results

Overall Performance. We first validate whether one adaptive model can outperform a family of compressors each trained for a single uniform ratio. Tab. 1 reports the comparison at 64×, with ϵ chosen so that MANTA's nominal ratio is strictly higher, ruling out any advantage from compressing less. MANTA reaches 56.87 token-F1 in domain and 42.25 out of domain, ahead of the strongest baseline on either split, by 9.3% and 5.4% relative, and the same ordering holds in EM, so a single information-adaptive model beats every dedicated per-ratio compressor on both splits and metrics.

Higher Compression Ratios. To show the performance at different compression ratios, we employ the same model at different tolerances against baselines retrained at the nearest lower ratio. Tab. 2 shows that MANTA obtains substantial gains at every compression ratio, on both splits and in both metrics. Specifically, at 91× it improves in-domain token-F1 by 7.9% relative over the strongest baseline at 85×, and it still clears the best 64× baseline by 5.1% while compressing 1.42× harder. Fig. 3 plots the resulting curve, in which one MANTA model consistently outperforms the per-ratio family by changing the tolerance. In App. B, we alter the same model at uniform k from 16× to 512×, smoothly and monotonically and including untrained k between the trained ladder points, firmly showing that every Matryoshka prefix is a usable compressed representation.

Table 2: Main results at different compression ratios on full MRQA.

Method	Ratio	ID		OOD	
		F1	EM	F1	EM
Full-FT	1×	81.73	72.01	66.59	56.21
ICAE	64×	45.70	34.85	34.26	25.78
500xC	64×	51.04	40.25	37.02	28.94
SAC	64×	52.01	40.93	40.10	30.59
MANTA	**65×**	**56.87**	**45.17**	**42.25**	**32.92**
ICAE	85×	43.98	33.15	33.11	24.25
500xC	85×	50.37	39.45	36.93	28.24
SAC	85×	50.65	39.94	37.68	28.60
MANTA	**91×**	**54.65**	**43.45**	**39.94**	**30.73**
ICAE	128×	42.70	32.14	32.66	23.80
500xC	128×	49.46	38.82	36.08	27.30
SAC	128×	48.50	38.13	36.82	28.09
MANTA	**134×**	**51.92**	**41.16**	**37.73**	**29.17**

Longer Contexts. To examine the generalization ability to longer contexts, we validate one model at a single tolerance holding up on inputs several times longer than any it is trained on. Tab. 3 evaluates needle-QA from 4K to 32K tokens, over which MANTA leads the dedicated

Table 3: Long-context needle-QA in token-F1 with one MANTA model at ϵ=0.05.

Context	4K	8K	16K	32K
SAC (64×)	35.6	26.8	23.8	9.9
MANTA (65×)	**36.7**	**31.3**	**28.5**	**17.2**

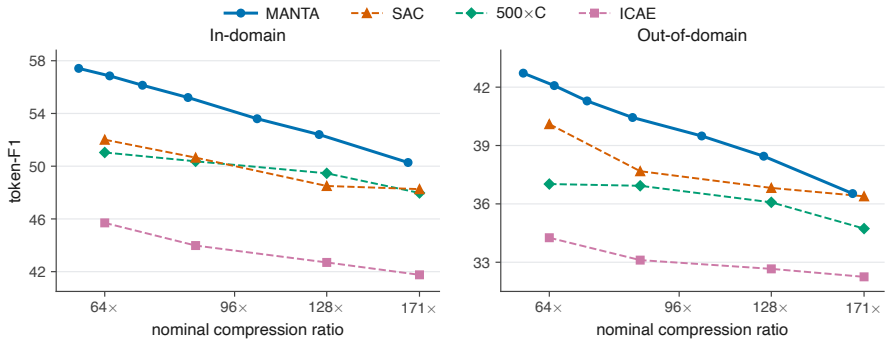

Figure 3: Performance curves of different compression ratios. Token-F1 is plotted against compression ratio on ID and OOD data, comparing one MANTA model with dedicated per-ratio baselines.

64× compressor at every length and by 18.3% relative on average. The margin widens monotonically with context length and is widest at 32K, where the baseline collapses to 9.9 token-F1 while MANTA retains 17.2. These results demonstrate that MANTA extrapolates robustly to context lengths far beyond its training range, without retraining.

Model Scales. We repeats the 64× comparison at 1B, 3B and 8B in Tab. 4 to validate the scalability of the method. It can be concluded that MANTA leads SAC at all three scales, by 8–9% relative in domain throughout, while out of domain the margin grows sharply with scale, from 5.4% at 1B to 16–17% at 3B and 8B. Measured against each scale's own uncompressed reference, MANTA recovers 70% of in-domain token-F1 at 1B and 85% at 8B, so a larger backbone gives up proportionally less to compression.

Table 4: Performance across model scales at 64× compression, evaluated by token-F1 on full MRQA.

Method	1B		3B		8B	
	ID	OOD	ID	OOD	ID	OOD
Full-FT	81.73	66.59	84.72	73.21	84.65	73.71
SAC	52.01	40.10	64.15	47.88	66.74	51.98
MANTA	**56.87**	**42.25**	**69.33**	**56.20**	**72.18**	**60.32**

Different Backbone Families. Tab. 5 repeats the comparison on Qwen2.5-1.5B at matched compression ratio to verify that the method is not tied to one backbone family. It is shown that MANTA leads at both compression ratios on both splits and in both metrics, by 5.9% relative in domain at 64× and by 7.8% out of domain at 128×, the latter margin wider than the corresponding one on Llama-1B. This greatly demonstrates the generalization of the proposed information-adaptive context compression.

Table 5: Generalization across backbone families on Qwen2.5-1.5B.

Method	Ratio	ID		OOD	
		F1	EM	F1	EM
SAC	64×	48.56	37.67	38.86	29.57
MANTA	64×	**51.42**	**39.83**	**40.73**	**31.40**
SAC	128×	44.64	34.47	35.55	27.12
MANTA	128×	**46.91**	**36.28**	**38.31**	**29.59**

4.3 Ablation Studies

Component Ablation. MANTA adds two components to a compressive encoder, so Tab. 6 removes each in turn under one recipe. Specifically, without the allocator the model is pure uniform-k Matryoshka. Nesting alone lifts in-domain token-F1 by 1.5% but leaves the out-of-domain split essentially unchanged; restoring the allocator adds a further 6.3% relative in domain and recovers the out-of-domain gain. Both components help in domain, but only the allocator moves the out-of-domain split.

Uniform vs. Adaptive Allocation. We evaluate one model adaptively against its own uniform-k prefixes to show the effectiveness of adaptive context compression. Each row of Tab. 7 pairs an adaptive operating point with a uniform truncation of the same model at a comparable mean allocation, so the two arms differ only in how the same nested representation is spent. Adaptive allocation wins on both splits in all three pairs. Concretely, a mean allocation of 7.84, itself below the uniform k=8 it is compared against, gains 3.3% in domain and 4.8% out of domain, and the top

Table 6: Architecture ablation at 64× compression on full MRQA.

Components		ID		OOD	
Matryoshka	Allocator	F1	EM	F1	EM
✗	✗	52.01	40.93	40.10	30.59
✓	✗	53.50	42.41	39.66	30.58
✓	✓	**56.87**	**45.17**	**42.25**	**32.92**

Table 7: Adaptive vs. uniform allocation of the model, evaluated by token-F1 on full MRQA.

mean k		ID		OOD	
Unif.	Adapt.	Unif.	Adapt.	Unif.	Adapt.
10	9.43	56.12	**57.83**	41.26	**43.19**
8	7.84	55.04	**56.87**	40.31	**42.25**
6	6.14	53.12	**55.21**	39.53	**40.44**

Table 8: Allocation-policy ablation at 64× compression under one recipe: no allocator, a naive global top-k head, and the ϵ-conditioned Elastic Allocator.

Policy	Allocation		ID		OOD	
	mean k	k std	F1	EM	F1	EM
None	8.00	0.00	53.50	42.41	39.66	30.58
Naive	7.87	0.72	54.60	43.30	40.70	31.51
Elastic	7.84	**2.41**	**56.87**	**45.17**	**42.25**	**32.92**

pair repeats this at 3.0% and 4.7% while again spending less than its uniform counterpart. Therefore, spending a given mean allocation unevenly beats spreading it uniformly without retraining models, showing its efficiency.

Influence of Different Allocation Policy. We employ a Elastic Allocation to achieve minimal sufficient allocation in MANTA. To figure out the influence of different allocation policy, we compare three policies under 64× compression in Tab. 8: no allocator (uniform k=8), a naive allocator applying the same head with a global top-k, and the ϵ-conditioned Elastic Allocator. The results indicate that the naive head brings little improvement over no allocator, whereas conditioning it on the tolerance gains a further 4.2% in domain and 3.8% out of domain at a smaller mean allocation. The assignment of Elastic Allocation also varies appreciably within a sample, with its standard deviation of k being 3.3× the naive one. The gain therefore comes strictly from content-driven differentiated allocation that the heterogeneity of §3.1 is not merely present but exploitable.

4.4 Further Analysis

Inference-Time Compression Control. To test whether ϵ is a continuous control rather than a memorized grid, we validate the performance of 13 values on full MRQA, 8 of them off the 5-point training grid. Fig. 4 shows mean allocation falling monotonically across different values, so that one model spans 52× to 342×, a 6.6× widening over which token-F1 declines by only about 20% relative on either split. The unseen tolerances fall on the trained curve, and ϵ=1.0, three times the largest trained value, still degrades gracefully. Since the ϵ scale is model-specific, a target ratio takes one calibration pass on held-out data, requiring no separate model for each ratio, which greatly shows the generalization ability. More details can be found in App. B.

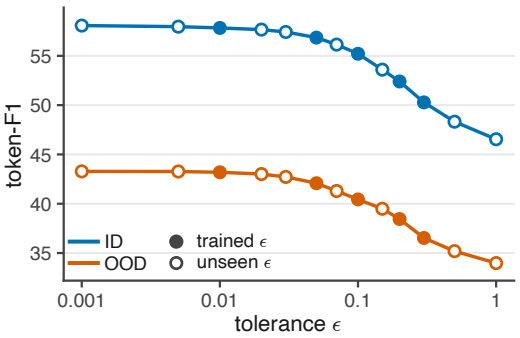

Figure 4: The tolerance ϵ as a continuous control. Five solid points are trained and eight hollow points are unseen during training.

Allocation Behavior. To certificate that specific tolerance can achieve adaptive allocation based on context, we fix ϵ=0.05 and record every per-chunk k on the full evaluation set. Fig. 5(a) aggregates them by dataset. It can be seen that per-dataset mean k stretches several-fold, and allocations within one sample span almost the whole ladder, ordered by content difficulty rather than by the ID/OOD split. Fig. 5(b) also illustrates one NaturalQuestions document, its four chunks in Matryoshka-token importance order with the cut at k. It vividly indicates that the answer-bearing chunk receives four times the allocation of a distractor. These results show that one tolerance differentiates across corpora and within an input following the context, showing the utility of adaptive context compression.

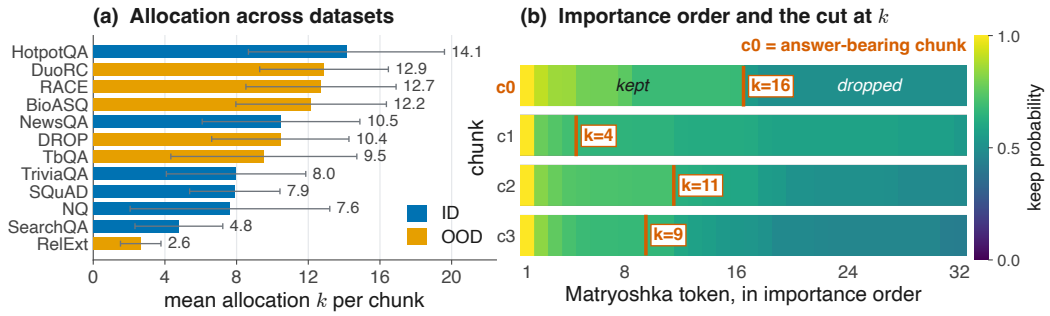

Figure 5: Allocation behavior. (a) Mean and per-chunk spread of k by dataset. (b) Four chunks of one NaturalQuestions document, their Matryoshka tokens in importance order and the cut at k.

Allocation Fidelity. An allocator that only fit its training domains would be of little use, so we validate whether the head still reproduces the minimial sufficient allocation of Eq. 6 off distribution, plotting per-dataset mean predicted k against oracle k on evaluation set. The twelve datasets illustratd in Fig. 6 follow one calibration relation at Pearson 0.83, the six OOD ones interleaving with the six ID ones rather than forming a displaced cluster. Where the predictor results in a deviation, it leans toward retention. The head therefore learned the oracle criterion, not the training corpora. More comparison results are shown in Tab. 14.

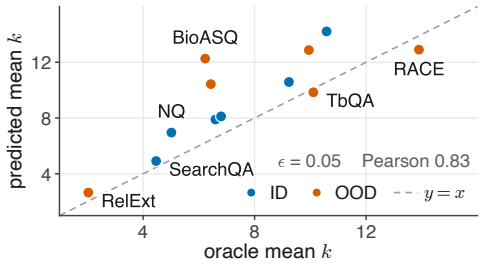

Figure 6: Allocation fidelity at $\epsilon = 0.05$ across ID and OOD datasets.

Representation Analysis. Differentiated allocation pays off only if the Matryoshka tokens kept carry distinct information, so we measure what nesting buys in the representation: the effective-rank ratio erank/k of the k retained states and the mean pairwise $|\cos|$ between them on the Value cache, against SAC at matched compression (Tab. 9). Here erank is the exponential of the entropy of the normalized singular values of the k retained states (Roy & Vetterli, 2007), so erank/k equals 1 only when they span k equally weighted directions, and $|\cos|$ is the mean absolute cosine similarity over all pairs of them; higher erank/k and lower $|\cos|$ both mean less redundancy. MANTA's retained states are less redundant at both ratios, with a higher effective rank and a correlation lower by 22.1% at 64×, and the order is front-loaded, one leading Matryoshka token already approaching the answer likelihood of all 32. Nesting therefore yields an ordered, low-redundancy code under which prefix truncation degrades gracefully.

Table 9: Redundancy of the retained Value states.

Method	erank$/k$ ↑		$\lvert\cos\rvert$ ↓	
	64×	128×	64×	128×
SAC	0.839	0.869	0.553	0.614
MANTA	**0.886**	**0.916**	**0.431**	**0.458**

5 Conclusion

In this paper, we introduce *information-adaptive context compression* to handle the problem of uniform context compression with fixed compression ratio. We formulate it as the *minimal sufficient allocation*: the smallest allocation whose excess loss stays under one global tolerance. Specifically, we introduce MANTA to realize it with (1) Matryoshka tokens, where arbitrary length of prefix is a valid compressed representation, and (2) a tolerance-conditioned Elastic Allocator that sets each chunk's allocation from content. Extensive experiments show that our method leads per-ratio baselines at different compression ratios, in and out of domain, on longer contexts, scales and backbones. Further analysis also verifies the effectiveness of adaptive allocation along information density.

AI USE STATEMENT

In this work, we used generative AI tools for writing assistance (drafting and polishing prose, and preparing the LaTeX source of tables and figures), for coding assistance (implementing parts of the training, evaluation, and analysis scripts), and for literature search support when assembling the related-work discussion. We have reviewed all AI-assisted work: AI-assisted code was read and tested by the authors, every number reported in the paper was checked against our experimental logs, and AI-suggested references were verified against their original sources. All research questions, method designs, experimental protocols, and scientific claims in this paper were formulated and validated by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. Matryoshka multimodal models. In *International Conference on Learning Representations (ICLR)*, 2025.

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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Guney, Volkan Cirik, and Kyunghyun Cho. SearchQA: A new Q&A dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*, 2017.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, 2019.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In *International Conference on Learning Representations (ICLR)*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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 (ICLR)*, 2022.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. Are you smarter than a sixth grader? Textbook question answering for multimodal machine comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural Questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics (TACL)*, 7:453–466, 2019.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL)*, 2017.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Zongqian Li, Yixuan Su, and Nigel Collier. 500xCompressor: Generalized prompt compression for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

Bokai Lin, Zihao Zeng, Zipeng Xiao, Siqi Kou, Tianqi Hou, Xiaofeng Gao, Hao Zhang, and Zhijie Deng. MatryoshkaKV: Adaptive KV compression via trainable orthogonal projection. In *International Conference on Learning Representations (ICLR)*, 2025.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics (TACL)*, 12:157–173, 2024a.

Tianyang Liu, Canwen Xu, and Julian McAuley. RepoBench: Benchmarking repository-level code auto-completion systems. In *International Conference on Learning Representations (ICLR)*, 2024b.

Xin Liu, Runsong Zhao, Pengcheng Huang, Xinyu Liu, Junyi Xiao, Chunyang Xiao, Tong Xiao, Shengxiang Gao, Zhengtao Yu, and Jingbo Zhu. Autoencoding-free context compression for LLMs via contextual semantic anchors. In *International Conference on Learning Representations (ICLR)*, 2026.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations (ICLR)*, 2020.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *Proceedings of the 15th European Signal Processing Conference (EUSIPCO)*, 2007.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. DuoRC: Towards complex language understanding with paraphrased reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

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 *International Conference on Learning Representations (ICLR)*, 2026.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, 2017.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R. Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. An overview of the BioASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16(1):138, 2015.

David Wingate, Mohammad Shoeybi, and Taylor Sorensen. Prompt compression and contrastive conditioning for controllability and toxicity reduction in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024.

Jing Xu, Arthur Szlam, and Jason Weston. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

An Yang, Baosong Yang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Howard Yen, Tianyu Gao, and Danqi Chen. Long-context language modeling with parallel context encoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Fengji Zhang, Bei Chen, Yue Zhang, Jacky Keung, Jin Liu, Daoguang Zan, Yi Mao, Jian-Guang Lou, and Weizhu Chen. RepoCoder: Repository-level code completion through iterative retrieval and generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023a.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Long context compression with Activation Beacon. In *International Conference on Learning Representations (ICLR)*, 2025.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H_2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023b.

Runsong Zhao, Xin Liu, Xinyu Liu, Pengcheng Huang, Chunyang Xiao, Tong Xiao, and Jingbo Zhu. Position IDs matter: An enhanced position layout for efficient context compression in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025.

APPENDIX

A MORE EXPERIMENTAL SETUP AND DETAILS

A.1 MORE TRAINING DETAILS

This section gives the settings behind the two-stage pipeline of §3.4, sketched in Fig. 7.

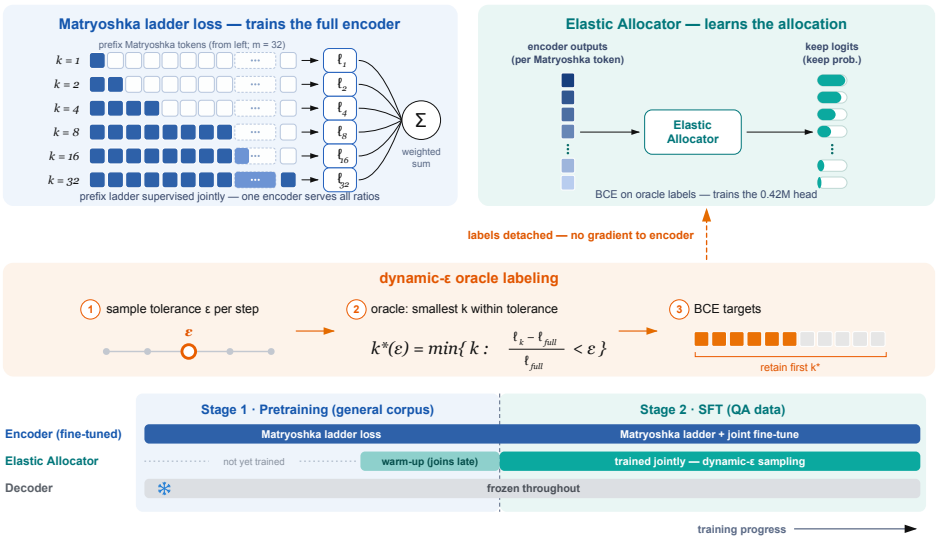

Figure 7: Two-stage training. Stage 1 learns the Matryoshka tokens on general-domain text and ramps the allocator in over its later part, and Stage 2 jointly trains the encoder and allocator on QA data with dynamic ϵ. The decoder remains frozen. In the top-left panel, the dashed cell abbreviates Matryoshka-token positions 9–31.

Stage 1 (pretraining) learns the Matryoshka tokens on general-domain text: from scratch, we train 20,000 steps on PG-19 (Rae et al., 2020) at batch 16 with learning rate 10^{-4} and 1,000 warmup steps, under the ladder objective of Eq. 3 over $K_{\text{set}} = \{1, 2, 4, 8, 16, 32\}$, resampling the tolerance at every step from the grid $\{0.01, 0.05, 0.1, 0.2, 0.3\}$. The allocator is skipped for 7,000 steps, then ramped in over 5,000.

Stage 2 (SFT) adapts encoder and allocator jointly to question answering: initialized from the Stage-1 adapter, it runs a further 20,000 steps on MRQA 320K at batch 16 with learning rate 5×10^{-5} and 1,000 warmup steps, with the allocator at warmup skip 0 and ramp 3,000.

A.2 EVALUATION DATA AND PROTOCOL

Data. Full-MRQA is the complete MRQA evaluation pool of n=67,854 examples: 58,221 in domain from SearchQA, NaturalQuestions-Short, SQuAD, TriviaQA-web, HotpotQA, and NewsQA (Dunn et al., 2017; Kwiatkowski et al., 2019; Rajpurkar et al., 2016; Joshi et al., 2017; Yang et al., 2018; Trischler et al., 2017), and 9,633 out of domain from BioASQ, DROP, DuoRC.ParaphraseRC, RACE, RelationExtraction, and TextbookQA (Tsatsaronis et al., 2015; Dua et al., 2019; Saha et al., 2018; Lai et al., 2017; Levy et al., 2017; Kembhavi et al., 2017). The per-dataset sample counts behind Tab. 1 are SQuAD 10,507, NewsQA 4,212, TriviaQA 7,785, SearchQA 16,980, HotpotQA 5,901, NQ 12,836, BioASQ 1,504, DROP 1,503, DuoRC 1,501, RACE 674, RelExt 2,948, and TextbookQA 1,503. Long-context evaluation uses needle-QA at 4K, 8K, 16K, and 32K with 500 samples each.

Decoding and Scoring. Every method encodes the context in chunks of C=512 and decodes greedily under a fixed seed, generating at most 128 new tokens on MRQA and 64 on needle-QA. We score with SQuAD-normalized token-F1 and EM, taking the maximum over references, identically for ev-

ery method in the paper. The Qwen experiments run on a Qwen-tokenized copy of the evaluation set.

B More Experimental Results

Complete Result Tables. This section gives the detailed results in the main paper, adding the metrics, ratios and tolerance points. They report the performance of the same models under the same protocol. Specifically, Tab. 10 repeats the three operating points at $65\times$, $91\times$ and $134\times$ against the baselines retrained at the nearest lower ratio, and adds a fourth one at $180\times$, where the same model is queried at ϵ=0.34 against baselines retrained at $171\times$. Tab. 11 adds the metric of EM in addition to token-F1 at every context length over the same 500 long context needle-QA samples.

Table 10: Detailed results across compression ratios covering four compression ratios on full MRQA in both metrics.

Method	Ratio	ID		OOD	
		F1	EM	F1	EM
Full-FT	1×	81.73	72.01	66.59	56.21
ICAE	64×	45.70	34.85	34.26	25.78
500xC	64×	51.04	40.25	37.02	28.94
SAC	64×	52.01	40.93	40.10	30.59
MANTA (ϵ=0.05)	**65×**	**56.87**	**45.17**	**42.25**	**32.92**
ICAE	85×	43.98	33.15	33.11	24.25
500xC	85×	50.37	39.45	36.93	28.24
SAC	85×	50.65	39.94	37.68	28.60
MANTA (ϵ=0.12)	**91×**	**54.65**	**43.45**	**39.94**	**30.73**
ICAE	128×	42.70	32.14	32.66	23.80
500xC	128×	49.46	38.82	36.08	27.30
SAC	128×	48.50	38.13	36.82	28.09
MANTA (ϵ=0.22)	**134×**	**51.92**	**41.16**	**37.73**	**29.17**
ICAE	171×	41.76	30.79	32.25	22.98
500xC	171×	47.97	37.52	34.73	26.12
SAC	171×	48.26	37.93	36.39	27.70
MANTA (ϵ=0.34)	**180×**	**49.75**	**39.36**	**36.42**	**27.94**

Table 11: Detailed results on long-context needle-QA.

Method	4K		8K		16K		32K	
	F1	EM	F1	EM	F1	EM	F1	EM
SAC (64×)	35.6	28.6	26.8	21.6	23.8	19.6	9.9	6.6
MANTA (65×)	**36.7**	**30.6**	**31.3**	**25.4**	**28.5**	**23.6**	**17.2**	**13.0**

Influence of Uniform-k. One Matryoshka model is meant to serve every compression ratio by truncation alone, so whether quality degrades gracefully or collapses once the prefix is short remains unsolved. To this end, Tab. 12 truncates the same model to its first k Matryoshka tokens at both the 6 trained points and 4 untrained intermediate ones, and evaluates each on full MRQA. It is shown that across the entire compression ratio from $512\times$ at k=1 to $16\times$ at k=32, in-domain token-F1 rises monotonically from 44.96 to 60.04, and OOD improves from 32.80 to 45.22. There is no collapse, indicating that every prefix is a usable operating point. Every untrained k further lands between its two neighbouring points on all four metrics, so the nested representation is continuous in between the trained ones, which licenses the arbitrary output length of the Elastic Allocator.

More Results on Different Tolerance Points. Tolerance offers a continuous control, yet only five values are seen in training, so Tab. 13 evaluates 13 of them, 8 unseen, to check that the performance generalizes well to unseen ones. Mean allocation drops monotonically from 9.80 to 1.50 Matryoshka tokens, widening the ratio $6.6\times$ from $52\times$ to $342\times$, while in-domain token-F1 declines smoothly from 58.07 to 46.54, and the within-sample k standard deviation falls from 3.04 to 0.12. This clearly

Table 12: Influence of uniform-k compression. Every row is the same model truncated to its first k Matryoshka tokens, evaluated on full MRQA. The 6 values of k in the training ladder are marked, while the other 4 are unseen at training time.

k	in training ladder	Nominal ratio	token-F1		EM	
			ID	OOD	ID	OOD
1	✓	512×	44.96	32.80	35.00	24.90
2	✓	256×	47.57	35.49	37.31	27.21
3		171×	49.34	37.44	38.80	28.92
4	✓	128×	50.79	38.27	40.04	29.83
6		85×	53.12	39.53	42.06	30.70
8	✓	64×	55.04	40.31	43.74	31.17
12		43×	57.20	42.21	45.48	32.67
16	✓	32×	58.34	43.59	46.42	34.10
24		21×	59.59	44.59	47.50	34.88
32	✓	16×	**60.04**	**45.22**	**47.89**	**35.32**

validates the effectiveness that one MANTA model not only performs well on trained ratio but also exhibits strong generalization ability to unseen compression ratio with diminished deviation.

Table 13: Results on more compression ratios. The 5 tolerances in the training grid are marked and set in bold, while the other 8 are unseen at training time.

ϵ	in training grid	mean k	≈ratio	token-F1		EM		within-sample k std
				ID	OOD	ID	OOD	
0.001		9.80	52×	58.07	43.28	46.22	33.82	3.04
0.005		9.63	53×	57.96	43.27	46.09	33.79	3.00
0.01	✓	**9.43**	**54×**	**57.83**	**43.19**	**46.00**	**33.77**	**2.96**
0.02		9.03	57×	57.66	43.01	45.85	33.60	2.84
0.03		8.62	59×	57.42	42.72	45.67	33.28	2.71
0.05	✓	**7.84**	**65×**	**56.87**	**42.25**	**45.17**	**32.92**	**2.41**
0.07		7.10	72×	56.14	41.29	44.65	31.98	2.08
0.1	✓	**6.14**	**83×**	**55.21**	**40.44**	**43.89**	**31.19**	**1.69**
0.15		4.96	103×	53.60	39.49	42.53	30.53	1.19
0.2	✓	**4.09**	**125×**	**52.40**	**38.45**	**41.56**	**29.83**	**0.89**
0.3	✓	**3.10**	**165×**	**50.28**	**36.53**	**39.82**	**28.20**	**0.57**
0.5		2.19	234×	48.32	35.19	37.99	26.96	0.31
1.0		1.50	342×	46.54	33.98	36.31	25.93	0.12

C Further Analysis of Allocation and Representation

Elastic Allocator Cross-Domain Fidelity. Tab. 14 further aggregates the the allocator's fidelity over the six ID and the six OOD datasets at different tolerances. On the full evaluation set we recompute the oracle k of Eq. 6 for every chunk in the same forward pass and compare it against the predicted k. Because the decoder loss falls monotonically in k, the oracle criterion is met exactly when $k_{\text{pred}} \geq k_{\text{oracle}}$, so the comparison yields not only how large the allocation error is but which way it points.

It can be concluded that: (1) predicted and oracle allocations align closely on the trained domains, at Pearson 0.92–0.97 and a mean absolute error of 1.2–1.7, and the untrained domains cost further 0.6–1.0 error and 2–5% of constraint satisfaction rather than breaking the relation. (2) Correlation is the more tolerance-sensitive: it weakens on the OOD split as ϵ grows, because a larger tolerance compresses the oracle allocations into a narrow low range, shrinking the within-sample k spread of Tab. 13 by 43% from ϵ=0.01 to ϵ=0.1. This leaves less variation for correlation to track, while the absolute error over the same span does not grow. (3) Moreover, the error is also directional: 71–83% of chunks receive at least their oracle allocation, and that share rises with ϵ. This indicates

Table 14: Detailed allocation fidelity at different tolerances. Predicted and oracle k are compared per dataset, where "satisfied" denotes $\Pr[k_{\text{pred}} \geq k_{\text{oracle}}]$.

ϵ	Pearson r		Mean abs. error		Satisfied	
	ID	OOD	ID	OOD	ID	OOD
0.01	0.96	0.88	1.32	1.94	72.7%	70.8%
0.05	0.97	0.77	1.66	2.47	80.9%	77.3%
0.10	0.92	0.69	1.23	2.27	83.1%	78.6%

that the head leans toward retention precisely where compression is most aggressive, which is the safe direction to miss in when a chunk's requirement is uncertain.

Redundancy of Retained Matryoshka Tokens. To validate the redundancy, we compute each chunk's effective-rank ratio erank/k over its k retained states from the spectral entropy (Roy & Vetterli, 2007), where higher means less redundant, and the mean pairwise $|\cos|$ between its retained Matryoshka tokens, where lower means less redundant, comparing SAC's own compressed tokens against MANTA's Matryoshka tokens at uniform k and matched counts. Tab. 15 provides detailed results with a cross-chunk global variant of the effective rank. The results show that MANTA is more faithful at equal allocation, which can be attributed to the lower redundancy among the Matryoshka tokens it retains.

Table 15: Matryoshka-token redundancy compared at matched compression ratio.

Method	Ratio	erank$/k$ (V) ↑	$\lvert\cos\rvert$ (V) ↓	global erank (V) ↑
SAC	64×	0.839	0.553	0.848
MANTA	64×	**0.886**	**0.431**	**0.870**
SAC	85×	0.845	0.596	0.859
MANTA	85×	**0.899**	**0.438**	**0.882**
SAC	128×	0.869	0.614	0.881
MANTA	128×	**0.916**	**0.458**	**0.896**

Ordered Residual Coding. Redundancy is an unordered, symmetric property and whether the Matryoshka tokens form a meaningful sequence remained underexplored. Therefore, we push teacher-forced gold answers through the deployment compression path and measure answer-NLL under subsets of Matryoshka tokens, setting the learned canonical prefix order against reverse, shuffled and random-subset controls over 300 samples. Tab. 16 reports the results of different prefix length.

Table 16: Influence of Matryoshka-token order. Answer-NLL are computed over 300 samples, where lower is better.

Matryoshka tokens k	1	2	4	8	16	32
Reverse	5.03	4.25	3.70	3.25	2.79	1.01
Shuffled	4.42	3.89	3.20	2.54	1.72	1.02
Random subset	4.50	3.88	3.19	2.42	1.68	1.01
Canonical (learned order)	**1.33**	**1.26**	**1.22**	**1.14**	**1.08**	1.01

The canonical order behaves like a residual code, meaning it possesses considerable efficiency starting from the first element. Specifically, one canonical Matryoshka token already reaches an answer-NLL of 1.33, close to the 1.01 of the full 32, and tightens gently to 1.08 at k=16. By contrast, the same Matryoshka tokens in any other arrangement start at 5.03 reversed, 4.42 shuffled and 4.50 as a random subset, and none reaches 1.33 until all 32 are kept. Moreover, marginal gains are correspondingly front-loaded, large for the first Matryoshka token and near zero afterwards. This strongly shows that the order and not the set carries the structure.