# WHAT STILL MATTERS: COMPRESSING LONG CONTEXT VIA SET-CONDITIONED MARGINAL VALUE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Long-context inference is essential to many applications of large language models (LLMs), yet processing lengthy inputs remains computationally expensive and often suffers from substantial information redundancy. Existing context compression methods alleviate this burden by reducing input length, and recent approaches further assign non-uniform compression rates according to segment-wise importance. In this work, we argue that the value of preserving a segment at high fidelity is inherently *set-dependent*: a segment that appears highly relevant in isolation may contribute little once its evidence is already covered, while a less relevant but complementary segment can provide greater marginal utility. To capture this dependency, we introduce **`ReValue`**, a set-conditioned context compression framework that continually reassesses what still matters as the high-fidelity set evolves. **`ReValue`** first derives compact query-aware representations for context segments, and then constructs a Determinantal Point Process (DPP)-based set objective that jointly captures query-conditioned quality and multi-scale semantic overlap. Rather than relying on fixed segment-wise scores, each candidate segment is evaluated by its marginal contribution relative to the current high-fidelity set, allowing redundant evidence to be progressively discounted while complementary information remains valuable. The resulting hybrid context preserves full token-level representations for high-value segments and compact representations for the rest, further supported by a representation diversity regularizer. Experiments demonstrate consistent gains across backbones, compression ratios, and benchmarks, including up to 11.04 points in average F1 and 5.01× faster first-token latency.

## 1 INTRODUCTION

Real-world applications often require large language models (LLMs) [1, 2, 3] to reason over extensive contextual information, such as long documents [4, 5, 6, 7], large codebases [8, 9, 10], and multi-turn conversations [11, 12, 13]. However, long-context inference faces two key challenges: **i)** processing lengthy inputs incurs substantial computational and memory overhead due to the quadratic cost of self-attention and the linear growth of the KV cache [14, 15, 16]; and **ii)** real-world long contexts often contain considerable information redundancy [17, 18, 19], which can distract the model from task-relevant evidence and degrade downstream performance [20]. Context compression [21, 22, 23, 24, 25] has therefore emerged as a promising approach to long-context inference by reducing input length and mitigating the adverse effects of redundant content.

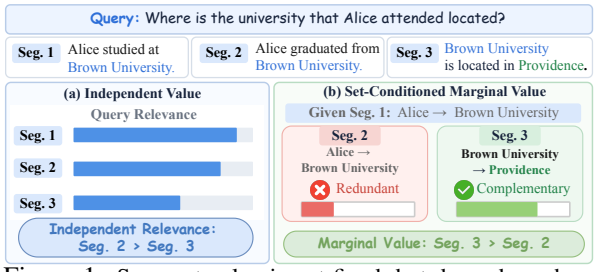



Figure 1: Segment value is not fixed, but depends on how much additional query-relevant information it provides beyond what is already available. Although Seg. 2 has higher independent relevance than Seg. 3, their relative values reverse once Seg. 1 is retained, as Seg. 3 provides complementary content.

Existing context compression methods can be broadly organized according to whether the compression process is conditioned on the query. **i) Query-agnostic** methods compress context without conditioning on the query, through strategies such as filtering less informative content based on intrinsic signals (*e.g.*, self-information or perplexity [18, 26, 27]) and distilling the context into



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.

a small set of latent representations that preserve their overall semantics [21, 23, 24, 25, 28, 29]. Although effective at reducing input length, these query-independent compression methods risk discarding or diluting query-relevant evidence, especially under aggressive compression [30]. **ii)** In contrast, **query-aware** methods incorporate the query into the compression process and prioritize task-relevant information through query-guided operations such as filtering, merging, or summarization [31, 32, 33]. Recent work takes a further step toward *adaptive compression*, assigning different compression rates or representation budgets across text segments based on various signals, such as query relevance [34], static maximum-similarity redundancy [35]. Such non-uniform allocation makes more effective use of a fixed representation budget, preserving task-relevant information at higher fidelity while more aggressively compressing content of lower utility to the current query.

Consider two segments (*e.g.*, Seg. 1 and Seg. 2 in Fig. 1) that are both highly relevant to the query but contain largely overlapping evidence. Although both may appear valuable when evaluated independently, once one is retained at full fidelity, the other provides little additional benefit. By contrast, a segment with lower independent relevance (*e.g.*, Seg. 3 in Fig. 1) that provides complementary evidence may offer greater marginal value. This observation suggests that the full-fidelity preservation value of a segment should be defined by its marginal contribution to the current high-fidelity set, rather than by a fixed segment-wise score.

Crucially, such a set-conditioned value jointly accounts for query relevance and complementarity under the current allocation, rather than encouraging semantic diversity alone. Accordingly, a semantically distinct but query-irrelevant segment remains of little value, whereas even a highly relevant segment becomes less valuable once its evidence has already been preserved at full fidelity. As the selected set expands, the marginal value of each remaining candidate must be re-evaluated: candidates that overlap with the retained evidence are progressively discounted, whereas those contributing complementary evidence remain valuable. This dynamic dependency leads to our central question: *Under a fixed high-fidelity budget, how can adaptive context compression continually reassess which segments still warrant full-fidelity preservation as the high-fidelity set evolves?*

To address this question, we propose `ReValue`, a set-conditioned context compression framework that continually re-evaluates what each segment still contributes as the high-fidelity set evolves. Given a query and a long context, `ReValue` first encodes all context segments in parallel and derives a compact query-aware representation for each segment. Based on these representations, **Multi-scale Set-Conditioned High-Fidelity Allocation** (§3.3) constructs a Determinantal Point Process kernel [36] that jointly models query-aware segment quality and multi-scale semantic overlap among segments, and greedily allocates high fidelity according to each candidate's marginal value relative to the current high-fidelity set. This marginal value favors query-relevant and complementary segments while discounting evidence already covered by the retained segments. The resulting allocation yields a hybrid compressed context: the selected high-fidelity segments retain their original token-level representations, whereas the remaining segments are represented via their compact representation rather than entirely discarded. To support reliable set-conditioned valuation, **Representation Diversity Regularization** (§3.4) further discourages excessive similarity among compact segment representations, providing a more discriminative representation space for high-fidelity allocation. Together, these designs allocate representation fidelity according to what still matters under the current allocation, rather than assigning each segment an independently estimated preservation score.

**Our contributions are threefold: (i)** We formulate query-aware adaptive context compression as a set-conditioned fidelity allocation problem, where each segment is evaluated by its marginal contribution relative to the evolving high-fidelity set rather than a fixed segment-wise score. **(ii)** We propose `ReValue`, a set-conditioned context compression framework that integrates query-conditioned segment quality and multi-scale semantic overlap into a Determinantal Point Process (DPP)-based objective to allocate high fidelity according to the marginal contribution of each segment relative to the current high-fidelity set. A representation diversity regularizer further discourages excessive similarity among compact segment representations, supporting more discriminative similarity estimates for set-conditioned high-fidelity allocation. **(iii)** Extensive experiments across multiple LLM backbones, compression ratios, and benchmarks demonstrate the effectiveness and efficiency of `ReValue`. Our method achieves higher task-relevant segment coverage than independent query-relevance-based allocation (§ 4.4), improves average F1 by up to 11.04 points over a strong query-aware adaptive compression baseline RAM [34], and delivers up to 5.01× faster time to first token with 50% lower peak GPU memory usage.

## 2 Related Work

**Query-Agnostic Long Context Compression.** Query-agnostic context compression reduces long inputs length without explicitly conditioning on a downstream query [18, 26, 27, 37]. Existing approaches can be broadly categorized into two groups. **i)** *Soft compression* encodes lengthy contexts into a small set of continuous representations that compactly preserve information from the original tokens [21, 24, 25, 38]. Existing methods construct compact representations by aggregating context into learned compression tokens [24, 28, 25, 39], compressing intermediate hidden states [23], or encoding text chunks with dedicated context encoders [29, 38, 40]. **ii)** A complementary paradigm, *hard compression*, directly shortens the original context by selecting or pruning textual content [18, 26, 15]. Representative methods estimate token informativeness to remove redundant content [26], formulate compression as learned token-level retention [27, 41], or dynamically revise pruning decisions according to attention and information-entropy signals during compression [42]. However, these query-agnostic paradigms may retain information irrelevant to the query while losing information needed to answer it, motivating query-aware compression that prioritizes task-relevant information.

**Query-Aware Long Context Compression.** Query-aware methods leverage query relevance to prioritize task-relevant information during compression [43, 31, 30, 44, 45, 46, 47, 48]. Recognizing that different context segments contribute unequally to the query [49, 50, 51, 52], recent methods further explore *query-aware adaptive compression*, allocating more representation capacity to important segments while compressing less relevant ones more aggressively [53, 54, 55]. For example, RAM varies representation fidelity according to query relevance [34], while COMI further adjusts this utility by penalizing each segment according to its maximum similarity to other context segments [35]. Despite this progress, how the value of each segment changes with the representation budget allocated to other segments remains underexplored. We explicitly model this dependency by formulating long-context compression as a set-conditioned fidelity allocation problem.

## 3 Method

### 3.1 Overview

We propose `ReValue` (Fig. 2), a set-conditioned context compression framework that continually re-evaluates each segment's preservation value according to its marginal contribution to the evolving high-fidelity set, rather than relying on a fixed segment-wise score. Given a query sequence $x_q$ and a long context divided into $N$ segments, `ReValue` constructs a query-aware hybrid context representation by distributing a fixed high-fidelity budget across the context segments. First, **Parallel Context Encoding** (Sec. 3.2) encodes the query and all segments in parallel, and aggregates the token-level features of each segment into a compact query-aware representation. Based on these representations, **Multi-Scale Set-Conditioned Hybrid Context Compression** (Sec. 3.3) constructs a Determinantal Point Process (DPP) kernel that combines query-conditioned quality with multi-scale inter-segment similarity, and greedily allocates high fidelity according to each candidate's marginal value relative to the current high-fidelity set. Consequently, the preservation value of a segment depends on the set already represented at full fidelity rather than on a fixed segment-wise score. The selected segments retain their original token-level representations, while the remaining segments use their compact representations, forming the final hybrid compressed context. Finally, `ReValue` introduces a **representation diversity regularization** that encourages informative distinctions among compact segment representations during training (Sec. 3.4).

### 3.2 Parallel Context Encoding

Given a query sequence $x_q$ and a long context divided into $N$ segments $\{x_i\}_{i=1}^{N}$, we encode the query and all context segments in parallel using a shared text encoder, yielding their token-level representations $\mathbf{H}_q$ and $\{\mathbf{H}_i\}_{i=1}^{N}$. For segment $x_i$, let $L_i$ denote its token length and $\mathbf{h}_{i,t}$ the $t$-th token representation in $\mathbf{H}_i$, and we derive a global query representation $\mathbf{q}$ by mean pooling over $\mathbf{H}_q$. For each segment $x_i$, we then aggregate its token representations according to their cosine similarity with $\mathbf{q}$, yielding a query-aware compact representation $\mathbf{c}_i \in \mathbb{R}^d$:

$$\mathbf{c}_i = \sum_{t=1}^{L_i} \alpha_{i,t}\mathbf{h}_{i,t}, \qquad \alpha_{i,t} = \frac{\exp(\cos(\mathbf{h}_{i,t}, \mathbf{q}))}{\sum_{t'=1}^{L_i} \exp(\cos(\mathbf{h}_{i,t'}, \mathbf{q}))}. \tag{1}$$

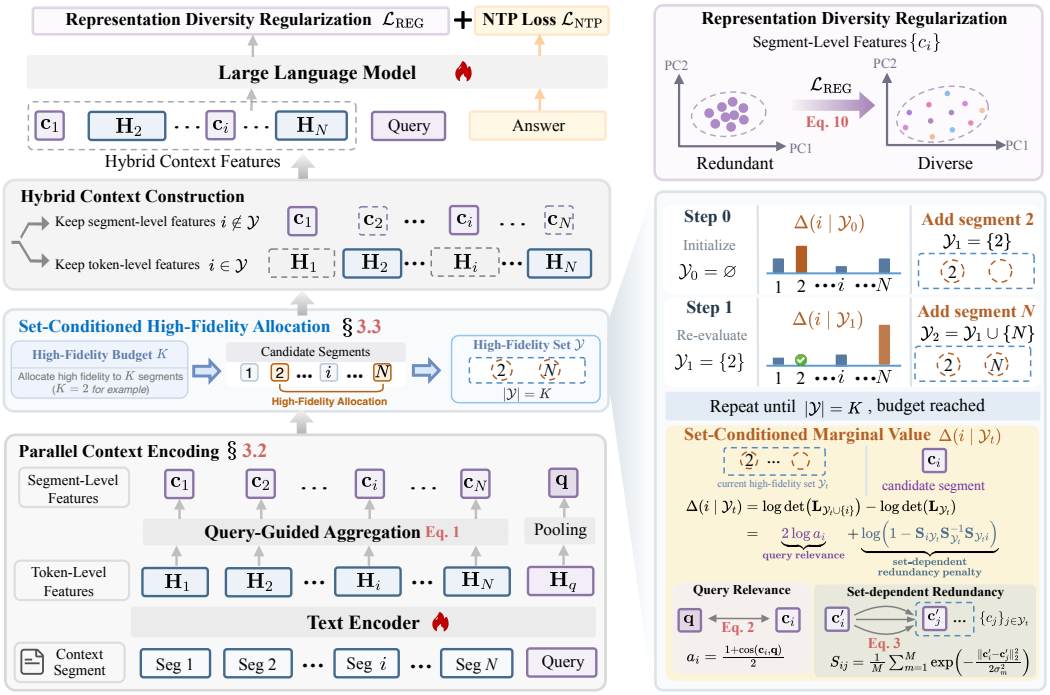



Figure 2: **Overview of `ReValue`.** Given query and long context, `ReValue` encodes segments in parallel to obtain token-level features $\{\mathbf{H}_i\}_{i=1}^N$ and compact query-aware features $\{\mathbf{c}_i\}_{i=1}^N$ (§ 3.2). Based on these compact features, `ReValue` progressively constructs the high-fidelity set $\mathcal{Y}$ under a budget of $K$ segments by selecting the candidate with the highest set-conditioned marginal value at each step (§ 3.3). This value jointly accounts for query relevance and set-dependent redundancy modeled through multi-scale semantic similarity. Segments in high-fidelity set $\mathcal{Y}$ retain token-level features $\mathbf{H}_i$, while the remaining segments use compact features $\mathbf{c}_i$, forming a hybrid context for downstream generation. A representation diversity regularizer discourages overly similar compact segment representations, preserving meaningful distinctions among context segments (§ 3.4).

The resulting $\{\mathbf{c}_i\}_{i=1}^N$ provide compact segment representations for the subsequent multi-scale set-conditioned hybrid context compression stage.

### 3.3 Multi-Scale Set-Conditioned Hybrid Context Compression

Given a target compression ratio $\rho$, `ReValue` sets the high-fidelity budget as $K = \left\lfloor \frac{NL_{\text{seg}}/\rho - N}{L_{\text{seg}} - 1} \right\rfloor$, where $L_{\text{seg}}$ is the segment length. The selected $K$ segments retain their full token-level representations, while the remaining segments are represented by compact vectors $\mathbf{c}_i$. Effective allocation should preserve query-relevant evidence without repeatedly assigning high fidelity to semantically overlapping content. Therefore, we formulate high-fidelity allocation as a set-conditioned problem, where the value of a segment depends jointly on its relevance to the query and the information it contributes beyond the current high-fidelity set. As the high-fidelity set evolves, this additional value changes: segments overlapping with retained information become less valuable, whereas complementary segments retain a larger marginal value. This formulation enables `ReValue` to continually re-evaluate what still matters during allocation, rather than relying on fixed segment-wise scores.

**Multi-Scale Context Valuation.** To determine which context segments should be retained at full fidelity, we characterize each segment from two complementary perspectives: its relevance to the query and its semantic overlap with the other segments. The former reflects query-specific importance of a segment, while the latter enables its additional value to be assessed relative to a selected set. Drawing on the quality–diversity formulation of Determinantal Point Processes (DPPs) [36], we jointly model these factors to capture redundancy at different semantic scales and evaluate each segment by its marginal contribution beyond the selected set.

For each segment $x_i$, we first measure its query relevance using its compact representation $\mathbf{c}_i$:

$$r_i = \frac{1 + \cos(\mathbf{c}_i, \mathbf{q})}{2}, \qquad a_i = \epsilon + (1 - \epsilon) r_i, \tag{2}$$

where $r_i \in [0, 1]$ denotes the query-relevance score and $a_i \in [\epsilon, 1]$ its corresponding quality weight. We set $\epsilon = 0.01$ to keep all quality weights strictly positive, preventing segments from receiving zero quality solely due to their query-relevance scores.

Though $a_i$ captures the query-conditioned quality of each segment, high-fidelity allocation also needs to account for semantic overlap among segments. Long contexts often contain redundancy of varying strengths, from near-duplicate content to partial semantic overlap. A single similarity scale may either overlook weaker redundancy or over-penalize segments that still provide complementary information. To obtain a more robust estimate of inter-segment redundancy, we first normalize each compact representation as $\mathbf{c}'_i = \mathbf{c}_i / \|\mathbf{c}_i\|_2$, and compute Gaussian similarities under multiple bandwidths:

$$S_{ij} = \frac{1}{M} \sum_{m=1}^{M} \kappa_{ij}^{(m)}, \qquad \kappa_{ij}^{(m)} = \exp\left( -\frac{\|\mathbf{c}'_i - \mathbf{c}'_j\|_2^2}{2\sigma_m^2} \right). \tag{3}$$

Here, a smaller bandwidth $\sigma_m$ responds primarily to close semantic matches, whereas a larger bandwidth captures overlap over a broader semantic neighborhood. Averaging across bandwidths yields a more robust estimate of inter-segment overlap without relying on a single similarity scale. `ReValue` then combines query-aware qualities and inter-segment similarities to form the DPP kernel:

$$\mathbf{L} = \mathbf{ASA}, \qquad \mathbf{A} = \mathrm{Diag}(a_1, \ldots, a_N), \tag{4}$$

where $\mathbf{A}$ encodes the query-aware quality of individual segments, while $\mathbf{S}$ characterizes semantic overlap among them. This provides the basis for the set-conditioned high-fidelity allocation described next.

**Set-Conditioned High-Fidelity Allocation.** Given a high-fidelity budget $K$, `ReValue` determines which $K$ segments retain their full token-level representations through the following set objective:

$$\mathcal{Y}^* = \arg \max_{\substack{\mathcal{Y} \subseteq \{1, \ldots, N\} \\ |\mathcal{Y}| = K}} \log\det(\mathbf{L}_\mathcal{Y}), \tag{5}$$

where $\mathbf{L}_\mathcal{Y}$ denotes the principal submatrix of $\mathbf{L}$ indexed by $\mathcal{Y}$. Since $\mathbf{L}_\mathcal{Y} = \mathbf{A}_\mathcal{Y} \mathbf{S}_\mathcal{Y} \mathbf{A}_\mathcal{Y}$, we define the corresponding set score as

$$F(\mathcal{Y}) = \log\det(\mathbf{L}_\mathcal{Y}) = 2 \sum_{i \in \mathcal{Y}} \log a_i + \log\det(\mathbf{S}_\mathcal{Y}). \tag{6}$$

The first term favors segments with high query-conditioned quality, whereas the second favors sets with lower inter-segment redundancy. Together, they encourage the limited high-fidelity budget to be allocated to query-relevant yet mutually complementary content. `ReValue` constructs $\mathcal{Y}$ greedily from the empty set. At each step, we evaluate a candidate segment $i \notin \mathcal{Y}$ by the change in the log-determinant set score upon adding it to the current high-fidelity set. Since the similarity matrix satisfies $S_{ii} = 1$, this marginal gain can be written as:

$$\begin{aligned} \Delta(i \mid \mathcal{Y}) &= F(\mathcal{Y} \cup \{i\}) - F(\mathcal{Y}) \\ &= \log\det(\mathbf{L}_{\mathcal{Y} \cup \{i\}}) - \log\det(\mathbf{L}_\mathcal{Y}) \\ &= \underbrace{2 \log a_i}_{\text{query-conditioned quality}} + \underbrace{\log\left(1 - \mathbf{S}_{i\mathcal{Y}} \mathbf{S}_\mathcal{Y}^{-1} \mathbf{S}_{\mathcal{Y}i}\right)}_{\text{set-dependent redundancy penalty}}. \end{aligned} \tag{7}$$

Here, $\Delta(i \mid \mathcal{Y})$ denotes the incremental log-determinant score under the fixed-cardinality allocation objective. Since $a_i \leq 1$ and the Schur complement does not exceed one, this score is non-positive for a positive-definite selected submatrix. Eq. 7 makes the set-conditioned nature of the allocation explicit. The first term reflects the query-conditioned quality of candidate $i$, whereas the second depends on its semantic relation to the current high-fidelity set $\mathcal{Y}$. As $\mathcal{Y}$ evolves, this set-dependent term changes: a candidate whose information is already well represented by the current set receives a stronger redundancy penalty, whereas a less redundant candidate retains a larger marginal value. We therefore add the candidate with the largest marginal value at each step and repeat this process until $|\mathcal{Y}| = K$. In this way, `ReValue` continually re-evaluates what each segment still contributes during allocation, rather than assigning it a fixed preservation value. Since selection is performed over $N$ context segments rather than individual tokens, incremental Cholesky updates yield an $\mathcal{O}(NK^2)$ greedy selection procedure. Empirically, set-conditioned allocation incurs only a small computational overhead, requiring less than 60 ms for inputs of up to 64K tokens, as detailed in Appendix D.

Table 1: Comparison with representative context compression methods under $4\times$ and $8\times$ compression.

| Methods | NaturalQA | | HotpotQA | | 2WikiMQA | | NaQA | News | AVG | |
|---|---|---|---|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 | F1 | F1 | EM | F1 |
| **Qwen3-4B-Instruct** | | | | | | | | | | |
| Original Prompt | 32.77 | 44.44 | 44.33 | 60.47 | 32.94 | 43.60 | 20.06 | 32.09 | 36.68 | 40.13 |
| *4× Compression Constraint* | | | | | | | | | | |
| ICAE | 17.93 | 19.95 | 17.97 | 25.84 | 27.13 | 30.80 | 10.77 | 20.58 | 21.01 | 21.59 |
| LLMLingua-2 | 23.09 | 35.72 | 28.20 | 41.02 | 25.17 | 31.58 | 19.03 | 29.56 | 25.49 | 31.38 |
| Provence | 31.11 | 43.39 | 42.38 | 56.11 | 39.52 | 48.32 | 24.46 | 28.30 | 37.67 | 40.12 |
| LongLLMLingua | 40.23 | 53.01 | 30.13 | 42.86 | 26.81 | 32.68 | 12.20 | 24.15 | 32.39 | 32.98 |
| COMI | 43.16 | 44.58 | 30.56 | 42.37 | 36.06 | 42.12 | 23.64 | 20.49 | 36.59 | 34.64 |
| RAM | 56.23 | 52.42 | 39.94 | 51.74 | 40.48 | 45.35 | 25.00 | 23.29 | 45.55 | 39.56 |
| `ReValue` | **61.51** | **56.56** | **47.49** | **60.07** | **50.26** | **56.10** | **36.76** | **38.45** | **53.09** | **49.59** |
| *8× Compression Constraint* | | | | | | | | | | |
| ICAE | 19.81 | 21.50 | 17.52 | 25.48 | 26.84 | 30.54 | 10.75 | 15.58 | 21.39 | 20.77 |
| LLMLingua-2 | 15.37 | 26.67 | 18.60 | 28.23 | 21.37 | 25.57 | 15.99 | 26.41 | 18.45 | 24.57 |
| Provence | 31.30 | 43.01 | 36.74 | 48.71 | 37.25 | 44.72 | 24.34 | 24.47 | 35.10 | 37.05 |
| LongLLMLingua | 31.41 | 45.27 | 24.45 | 35.51 | 23.65 | 27.91 | 9.76 | 20.48 | 26.50 | 27.79 |
| COMI | 32.24 | 36.38 | 20.38 | 33.35 | 30.58 | 34.12 | 18.89 | 16.52 | 27.73 | 27.85 |
| RAM | 48.09 | 45.89 | 33.15 | 43.91 | 33.64 | 38.18 | 21.09 | 18.27 | 38.29 | 33.47 |
| `ReValue` | **59.40** | **54.60** | **38.20** | **50.40** | **40.20** | **49.44** | **33.16** | **34.93** | **45.93** | **44.51** |
| **LLaMA-3.1-8B-Instruct** | | | | | | | | | | |
| Original Prompt | 37.63 | 48.25 | 26.81 | 45.86 | 26.07 | 39.60 | 19.26 | 34.79 | 30.17 | 37.55 |
| *4× Compression Constraint* | | | | | | | | | | |
| ICAE | 21.36 | 23.12 | 19.43 | 27.87 | 28.98 | 32.71 | 9.25 | 23.30 | 23.26 | 23.25 |
| LLMLingua-2 | 29.98 | 43.92 | 29.83 | 45.25 | 18.16 | 32.76 | 24.93 | 30.08 | 25.99 | 35.39 |
| Activation Beacon | 47.04 | **58.97** | 43.78 | 57.44 | 36.69 | 44.20 | 28.29 | 28.52 | 42.50 | 43.48 |
| Provence | 37.25 | 49.13 | 40.57 | 57.51 | 30.25 | 45.87 | 30.11 | 30.14 | 36.02 | 42.55 |
| LongLLMLingua | 45.50 | 58.34 | 34.88 | 51.40 | 24.58 | 35.30 | 19.17 | 26.32 | 34.99 | 38.11 |
| RAM | 62.90 | 57.06 | 41.64 | 54.89 | 43.24 | 49.06 | 30.63 | 30.25 | 49.26 | 44.38 |
| `ReValue` | **64.82** | **58.97** | **48.25** | **61.30** | **50.95** | **56.83** | **35.18** | **32.93** | **54.67** | **49.04** |
| *8× Compression Constraint* | | | | | | | | | | |
| ICAE | 21.77 | 24.04 | 19.51 | 28.17 | 29.54 | 33.40 | 10.97 | 23.68 | 23.61 | 24.05 |
| LLMLingua-2 | 20.11 | 33.58 | 21.29 | 34.55 | 14.05 | 27.62 | 22.46 | 27.55 | 18.48 | 29.15 |
| Activation Beacon | 41.21 | 55.09 | 36.29 | 48.57 | 35.53 | 42.98 | 26.80 | 25.19 | 37.68 | 39.73 |
| Provence | 34.16 | 47.19 | 37.22 | 49.91 | 28.81 | 43.12 | 29.15 | 26.19 | 33.40 | 39.11 |
| LongLLMLingua | 39.50 | 54.30 | 28.63 | 44.10 | 20.33 | 30.13 | 15.84 | 21.56 | 29.49 | 33.19 |
| RAM | 60.33 | 54.81 | 33.09 | 45.61 | 33.44 | 38.76 | 30.43 | 26.01 | 42.29 | 39.12 |
| `ReValue` | **65.56** | **57.35** | **42.08** | **54.42** | **43.12** | **48.34** | **33.60** | **29.74** | **50.25** | **44.69** |

**Hybrid Context Construction.** The selected set $\mathcal{Y}$ determines the representation fidelity of each segment rather than whether it is retained. Segments in $\mathcal{Y}$ preserve their full token-level representations, while the remaining segments use their compact representations. Thus, `ReValue` constructs its final representation:

$$\mathbf{R}_i = \begin{cases} \mathbf{H}_i, & i \in \mathcal{Y}, \\ \mathbf{c}_i, & i \notin \mathcal{Y}. \end{cases} \tag{8}$$

Finally, the segment representations are restored to their original context order and concatenated to form the hybrid compressed context $\mathbf{R}$ for downstream generation.

### 3.4 Representation Diversity Regularization

Set-conditioned high-fidelity allocation relies on compact segment representations to estimate query-conditioned quality and inter-segment similarity. When these representations become overly similar, the resulting similarity kernel may provide insufficient discrimination among candidate segments, limiting the effectiveness of high-fidelity allocation. To alleviate this issue, we introduce a representation diversity regularizer that discourages excessive similarity among compact segment representations during training. Specifically, we apply the same determinantal formulation with detached query-conditioned quality weights:

$$\widetilde{\mathbf{A}} = \mathrm{Diag}(\mathrm{sg}(a_1), \ldots, \mathrm{sg}(a_N)), \qquad \widetilde{\mathbf{L}} = \widetilde{\mathbf{A}}\mathbf{S}\widetilde{\mathbf{A}}, \tag{9}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operation. The regularization objective is

$$\mathcal{L}_{\mathrm{REG}} = -\frac{1}{N} \log\det\left(\mathbf{I}_N + \widetilde{\mathbf{L}}\right). \tag{10}$$

The log-determinant discourages excessive similarity among query-weighted compact segment representations, while the identity matrix ensures that the objective remains well-defined when the kernel is rank-deficient. We detach the quality weights so that the regularization loss does not propagate gradients through the quality-weight factors. Instead, gradients propagate through the similarity kernel $\mathbf{S}$ to the compact segment representations and the shared encoder. This regularization is intended to

Table 2: Performance comparison on LongBench under $4\times$ compression. "Original Prompt" denotes directly feeding the full context to the backbone LLM.

| Method | Multi-doc QA | | Single-Doc QA | | Summarization | | | Few-shot | | Code | | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MuSiQue | DuReader | MFieldQA | Qasper | GReport | QMSum | VCSUM | SAMSum | TREC | LCC | ReBench | |
| Original Prompt | 5.65 | 27.47 | 26.22 | 16.76 | 28.25 | 18.86 | 11.41 | 45.09 | 73.50 | 55.21 | 50.43 | 32.62 |
| COMI | 26.61 | 10.28 | 27.94 | 22.52 | 13.78 | 8.77 | 11.21 | 16.06 | 38.00 | 25.95 | 20.44 | 20.14 |
| RAM | 33.42 | 13.82 | **36.35** | 23.18 | 9.56 | 8.51 | 12.77 | 22.23 | 48.50 | 25.19 | 21.65 | 23.20 |
| `ReValue` | **37.96** | **17.10** | 35.89 | **28.59** | **21.92** | **10.71** | **15.11** | **32.00** | **57.00** | **34.14** | **22.88** | **28.48** |

provide a more discriminative similarity kernel for set-conditioned high-fidelity allocation. We use the standard next-token prediction loss $\mathcal{L}_{\text{NTP}}$ to supervise autoregressive generation. Combining it with the proposed representation diversity regularizer gives the overall training objective:

$$\mathcal{L} = \mathcal{L}_{\text{NTP}} + \lambda_{\text{REG}} \mathcal{L}_{\text{REG}}, \tag{11}$$

where $\lambda_{\text{REG}}$ controls the strength of the representation diversity regularization.

# 4 Experiment

## 4.1 Experimental Setup

**Training Dataset and Evaluation.** Following RAM [34], we construct a training set of 20,000 examples randomly sampled from HotpotQA [56], 2WikiMultiHopQA (2WikiMQA) [57], Natural Questions (NaturalQA) [58], NarrativeQA (NaQA) [59], and Multi-News (News) [60]. The maximum input length is set to 20K tokens during training. We evaluate `ReValue` in both in-domain and out-of-domain settings. For in-domain evaluation, we use the held-out evaluation splits, and primarily report Exact Match (EM) and F1 scores [56, 34]. For out-of-domain evaluation, we evaluate on 11 benchmarks from LongBench [61], spanning multi-document question answering (MuSiQue and DuReader), single-document question answering (MFieldQA and Qasper), summarization (GReport, QMSum, and VCSUM), few-shot learning (SAMSum and TREC), and code completion (LCC and ReBench). Unless otherwise specified, all experiments are conducted under a $4\times$ compression ratio.

**Implementation Details.** Following RAM [34], the compressor encoder uses the first eight Transformer layers of the corresponding backbone and is initialized with their pretrained weights. We divide each context into non-overlapping segments of 50 tokens by default. The model is trained in bfloat16 precision with a global batch size of 24 and an initial learning rate of $(1 \times 10^{-5})$ under a linear learning-rate schedule. At each training step, we uniformly sample a compression ratio ($\rho$) from (2, 4, 8, 16, 32), allowing the model to accommodate different compression budgets.

**Competitors.** We compare `ReValue` with representative context compression methods, including query-agnostic approaches (ICAE [28], LLMLingua-2 [41], and Activation Beacon [23]), query-aware hard compression (LongLLMLingua [43] and Provence [32]), and query-aware adaptive soft or hybrid compression (COMI [35] and RAM [34]). We also include the uncompressed original context as a reference, denoted as "Original Prompt". Further details are provided in Appendix B.

## 4.2 Main Results

**In-domain Results.** Table 1 presents the main results under $4\times$ and $8\times$ compression using Qwen3-4B-Instruct [62] and LLaMA-3.1-8B-Instruct [63]. `ReValue` consistently matches or outperforms all competing compression methods across the reported metrics and settings, demonstrating its effectiveness across different compression ratios and model backbones.

**Out-of-domain Generalization.** We further evaluate `ReValue` on LongBench using Qwen3-4B-Instruct, covering diverse tasks including multi-document QA, single-document QA, summarization, few-shot learning, and code understanding, as shown in Table 2. `ReValue` achieves the highest average score among the compared compression methods, outperforming RAM and COMI by 5.28 and 8.34 points, respectively. These results indicate that the effectiveness of `ReValue` extends beyond the datasets used for training to diverse out-of-domain tasks.

**Performance across Different Compression Ratios.** Figure 5 compares `ReValue` with COMI and RAM on NaturalQA using Qwen3-4B-Instruct at compression ratios from $2\times$ to $32\times$. As

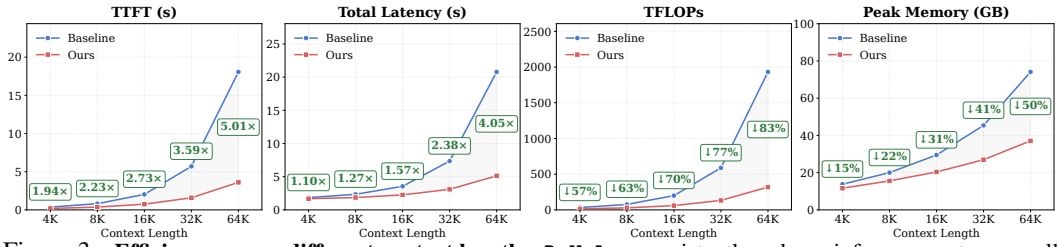



Figure 3: **Efficiency across different context lengths.** `ReValue` consistently reduces inference cost across all context lengths, with increasingly pronounced benefits as the context grows.

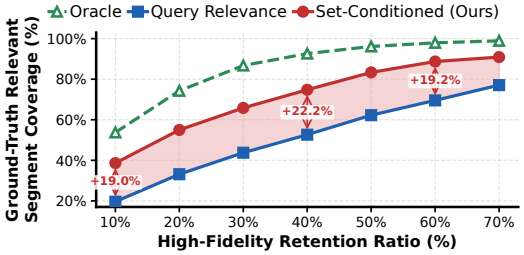



Figure 4: Coverage of task-relevant segments under different high-fidelity allocation strategies.

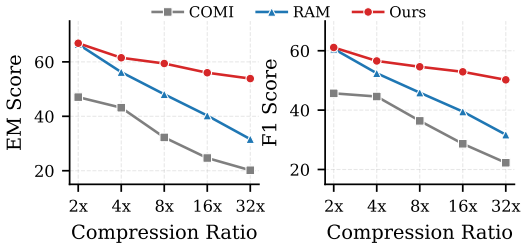



Figure 5: Performance under increasing compression ratios on NaturalQA [58].

compression becomes more aggressive, all methods experience performance degradation, whereas `ReValue` exhibits a substantially slower decline in both EM and F1. Notably, the performance gap between `ReValue` and RAM consistently widens as the compression ratio increases, indicating that the benefit of `ReValue` becomes more pronounced under tighter representation budgets.

**Efficiency Analysis.** We evaluate inference efficiency under $4\times$ compression using Qwen3-4B-Instruct [62] with a batch size of 1 and a maximum generation length of 64 tokens. We report time to first token (TTFT), total inference latency, computational cost in TFLOPs, and peak GPU memory usage. Specifically, TTFT is measured from receiving the original uncompressed input to generating the first output token, while total latency covers the entire end-to-end inference process, including context compression, prefill, and decoding. As shown in Figure 3, `ReValue` consistently reduces inference overhead compared with the baseline, with increasingly pronounced gains as the context length grows. At 64K tokens, `ReValue` achieves a $5.01\times$ TTFT speedup, a $4.05\times$ end-to-end speedup, an $83\%$ reduction in TFLOPs, and a $50\%$ reduction in peak GPU memory.

### 4.3 Ablation Study

**Effect of Regularization Weight.** As shown in Table 4, removing the regularizer ($\lambda_{\text{REG}} = 0$) consistently degrades EM and F1 across all three QA benchmarks. Introducing representation diversity regularization improves downstream performance, indicating that discouraging excessive similarity among compact segment representations benefits the proposed compression framework. Increasing $\lambda_{\text{REG}}$ from 1 to 2 yields only marginal changes, suggesting that performance is relatively stable within the evaluated range. We therefore adopt $\lambda_{\text{REG}} = 1$ as the default setting.

**Generalization to Another LLM Backbone.** To examine whether the effectiveness of `ReValue` generalizes across different LLM backbones, we further conduct experiments with Qwen3-8B under $4\times$ and $8\times$ compression. As shown in Table 5, `ReValue` consistently outperforms RAM across all three QA benchmarks and both compression ratios. At $4\times$ compression, `ReValue` improves the average EM and F1 by 6.11 and 5.72 points, respectively. The gains become even more pronounced under the more aggressive $8\times$ compression, reaching 9.25 points in average EM and 9.30 points in average F1. These results demonstrate that the effectiveness of `ReValue` is not specific to a particular backbone and remains robust under stronger compression.

**Effect of Multi-Scale Gaussian Bandwidths.** Table 6 studies different choices of the Gaussian bandwidth set $\{\sigma_m\}_{m=1}^{M}$ in Eq. 3 under $4\times$ compression. Compared with direct similarity, using Gaussian similarities over multiple bandwidths consistently improves the average performance, demonstrating the benefit of modeling inter-segment redundancy at multiple similarity scales. Specifically, using two bandwidths $\{0.5, 1.0\}$ improves the average EM from 50.96 to 52.16, while $\{0.25, 0.5, 1.0\}$ further improves it to 53.09 and achieves the best overall performance. The alternative three-bandwidth

Table 3: Effect of high-fidelity allocation strategies.

| Allocation Strategy | 2× | 4× | 8× | 16× | 32× |
|---|---|---|---|---|---|
| Random | 46.18 | 32.81 | 26.29 | 21.47 | 19.21 |
| Query Relevance | 52.43 | 56.85 | 51.69 | 48.98 | 44.54 |
| Static Relevance–Redundancy | 59.96 | 57.09 | 51.37 | 48.68 | 45.95 |
| `ReValue` | **66.85** | **61.51** | **59.40** | **56.01** | **53.86** |

Table 4: Effect of the regularization weight $\lambda_{\mathrm{REG}}$.

| $\lambda_{\mathrm{REG}}$ | NaturalQA | | HotpotQA | | 2WikiMQA | |
|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 |
| 0 | 59.74 | 53.04 | 44.08 | 56.02 | 48.67 | 54.44 |
| 1 | 61.51 | 56.56 | **47.49** | **60.07** | 50.26 | **56.10** |
| 2 | **63.31** | **57.63** | 47.25 | 59.65 | **50.35** | **56.10** |

Table 5: Results on Qwen3-8B under $4\times$ and $8\times$ compression.

| Method | Comp. | NaturalQA | | HotpotQA | | 2WikiMQA | | Average | |
|---|---|---|---|---|---|---|---|---|---|
| | | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| RAM | 4× | 67.08 | 60.02 | 47.47 | 59.81 | 51.86 | 57.70 | 55.47 | 59.18 |
| `ReValue` | 4× | **72.45** | **67.83** | **55.96** | **66.68** | **56.33** | **60.18** | **61.58** | **64.90** |
| RAM | 8× | 61.66 | 56.46 | 38.71 | 49.84 | 41.63 | 46.82 | 47.33 | 51.04 |
| `ReValue` | 8× | **68.95** | **65.33** | **49.90** | **60.54** | **50.91** | **55.16** | **56.59** | **60.34** |

Table 6: Effect of Gaussian bandwidth set.

| Kernel Scales | NQ | HQA | MQA | Avg. |
|---|---|---|---|---|
| Direct similarity | 60.04 | 44.34 | 48.50 | 50.96 |
| $\{0.5, 1.0\}$ | 63.16 | 44.92 | 48.39 | 52.16 |
| $\{0.25, 0.5, 1.0\}$ | 61.51 | **47.49** | **50.26** | **53.09** |
| $\{0.5, 1.0, 1.5\}$ | 62.66 | 47.14 | 49.29 | 53.03 |
| $\{0.25, 0.5, 1.0, 1.5\}$ | **63.26** | 46.10 | 48.59 | 52.65 |

configuration $\{0.5, 1.0, 1.5\}$ achieves a comparable average EM of 53.03, suggesting that the improvement is not sensitive to a single specific bandwidth choice. Adding a fourth bandwidth does not yield further improvement, with the average EM decreasing to 52.65. We therefore use $\{0.25, 0.5, 1.0\}$ as the default bandwidth set in our experiments.

### 4.4 Effectiveness of Set-Conditioned Allocation

**Does set-conditioned allocation improve downstream performance?** We evaluate different high-fidelity allocation strategies on NaturalQA across compression ratios from $2\times$ to $32\times$, reporting Exact Match (EM). We compare `ReValue` with three inference-time alternatives while keeping all other components unchanged: *Random*, which randomly selects segments; *Query Relevance*, which prioritizes query-relevant segments; and *Static Relevance–Redundancy*, which ranks segments by subtracting their highest similarity to other context segments from their query-relevance scores. As shown in Table 3, static redundancy-aware scoring does not consistently improve upon query relevance alone, whereas `ReValue` achieves the highest EM across all compression ratios.

**Does set-conditioned allocation better cover task-relevant segments?** We examine whether the limited high-fidelity budget is effectively allocated to task-relevant evidence by measuring the coverage of ground-truth relevant segments at different retention ratios. For evaluation, we randomly sample a total of 1,000 held-out examples from 2WikiMQA [57], none of them are used for training. Ground-truth relevant segments are identified by mapping the annotated supporting facts to their corresponding context segments. Given ground-truth relevant segments $\mathcal{G}$ and selected high-fidelity set $\mathcal{Y}$, we compute $\mathrm{Coverage} = |\mathcal{Y} \cap \mathcal{G}|/|\mathcal{G}|$, which measures the fraction of relevant segments retained at full fidelity. As shown in Fig. 4, we compare our *Set-Conditioned Allocation* with *Query Relevance*, which independently ranks segments by their query relevance. We also report the *oracle* upper bound, corresponding to the maximum achievable coverage under each retention ratio. Our method consistently achieves higher coverage across all retention ratios. These results indicate that set-conditioned allocation directs more of the limited high-fidelity budget toward task-relevant segments than independent query-relevance-based allocation.

## 5 Conclusion

In this work, we introduced `ReValue`, a set-conditioned framework for long-context compression that rethinks segment importance as a dynamic marginal value rather than a fixed segment-wise score. By jointly considering query relevance and set-dependent redundancy, `ReValue` progressively prioritizes segments that are relevant to the query while providing complementary information relative to the current high-fidelity set. This is achieved through a DPP-based set objective that combines query-aware segment quality and multi-scale semantic similarity, together with a hybrid representation scheme and representation diversity regularization. Extensive experiments demonstrate that set-conditioned allocation improves downstream task performance and achieves higher coverage of task-relevant segments under limited high-fidelity budgets.

## AI USE STATEMENT

In this work, we used generative AI tools to assist with language proofreading, improve the clarity of the manuscript, and support the retrieval and discovery of related work. Generative AI tools were also used to assist with parts of the coding process. We did not use generative AI tools to generate experimental results, perform data analysis, or draw scientific conclusions. All AI-assisted content was carefully reviewed and verified by the authors. We take full responsibility for the final content of this manuscript.

## REFERENCES


[1] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024. 1

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in neural information processing systems*, volume 33, pages 1877–1901, 2020. 1

[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[4] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 1

[5] Kimi Team, Yifan Bai, Yiping Bao, Y Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025. 1

[6] Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, Nanning Zheng, and Furu Wei. Longnet: Scaling transformers to 1,000,000,000 tokens. *arXiv preprint arXiv:2307.02486*, 2023. 1

[7] Aydar Bulatov, Yuri Kuratov, Yermek Kapushev, and Mikhail S Burtsev. Scaling transformer to 1m tokens and beyond with rmt. *arXiv preprint arXiv:2304.11062*, 2023. 1

[8] Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026. 1

[9] Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. Redpajama: an open dataset for training large language models. In *Advances in Neural Information Processing Systems*, 2024. 1

[10] Thomas Wolf, Loubna Ben Allal, Leandro von Werra, Li Jia, and Armel Zebaze. A dataset of python files from github, 2023. 1

[11] Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*, 2024. 1

[12] 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*, 2019. 1

[13] 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 (volume 1: long papers)*, pages 5180–5197, 2022. 1

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[15] 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*, volume 37, pages 22947–22970, 2024. 1, 3

[16] Xiang Li, Yiqun Yao, Xin Jiang, Xuezhi Fang, Chao Wang, Xinzhang Liu, Zihan Wang, Yu Zhao, Xin Wang, Yuyao Huang, et al. 52b to 1t: Lessons learned via tele-flm series. *arXiv preprint arXiv:2407.02783*, 2024. 1

[17] Jiajie Jin, Xiaoxi Li, Guanting Dong, Yuyao Zhang, Yutao Zhu, Yongkang Wu, Zhonghua Li, Ye Qi, and Zhicheng Dou. Hierarchical document refinement for long-context retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3502–3520, 2025. 1

[18] 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*, pages 6342–6353, 2023. 1, 3

[19] Hao Sun, Hengyi Cai, Yuchen Li, Xuanbo Fan, Xiaochi Wei, Shuaiqiang Wang, Yan Zhang, and Dawei Yin. Enhancing retrieval-augmented generation via evidence tree search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24116–24127, 2025. 1

[20] 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*, 12:157–173, 2024. 1

[21] 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*, pages 5621–5634, 2022. 1, 2, 3

[22] Alex Jinpeng Wang, Linjie Li, Yiqi Lin, Min Li, Lijuan Wang, and Mike Zheng Shou. Leveraging visual tokens for extended text contexts in multi-modal learning. In *Advances in Neural Information Processing Systems*, 2024. 1

[23] Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Soaring from 4k to 400k: Extending llm's context with activation beacon. *arXiv preprint arXiv:2401.03462*, 2024. 1, 2, 3, 7

[24] Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems*, volume 36, 2024. 1, 2, 3

[25] 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*, pages 3829–3846, 2023. 1, 2, 3

[26] 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*, pages 13358–13376, 2023. 1, 3

[27] Kolby Nottingham, Yasaman Razeghi, Kyungmin Kim, JB Lanier, Pierre Baldi, Roy Fox, and Sameer Singh. Selective perception: Learning concise state descriptions for language model actors. In *Association for Computational Linguistics*, pages 327–341, 2024. 1, 3

[28] 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 *International Conference on Learning Representations*, 2024. 2, 3, 7, 15

[29] Howard Yen, Tianyu Gao, and Danqi Chen. Long-context language modeling with parallel context encoding. In *Association for Computational Linguistics*, 2024. 2, 3

[30] Barys Liskavets, Maxim Ushakov, Shuvendu Roy, Mark Klibanov, Ali Etemad, and Shane K Luke. Prompt compression with context-aware sentence encoding for fast and improved llm inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24595–24604, 2025. 2, 3

[31] Zhiwei Cao, Qian Cao, Yu Lu, Ningxin Peng, Luyang Huang, Shanbo Cheng, and Jinsong Su. Retaining key information under high compression ratios: Query-guided compressor for llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12685–12695, 2024. 2, 3

[32] Nadezhda Chirkova, Thibault Formal, Vassilina Nikoulina, and Stéphane Clinchant. Provence: efficient and robust context pruning for retrieval-augmented generation. In *International Conference on Learning Representations*, volume 2025, pages 38082–38106, 2025. 2, 7

[33] Fangyuan Xu, Weijia Shi, and Eunsol Choi. Recomp: Improving retrieval-augmented lms with context compression and selective augmentation. In *International Conference on Learning Representations*, volume 2024, pages 43478–43502, 2024. 2

[34] Jiwei Tang, Shilei Liu, Zhicheng Zhang, Qingsong Lv, Runsong Zhao, Tingwei Lu, Langming Liu, Haibin Chen, Yujin Yuan, Hai-Tao Zheng, et al. Read as human: Compressing context via parallelizable close reading and skimming. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28393–28406, 2026. 2, 3, 7, 15

[35] Jiwei Tang, Shilei Liu, Zhicheng Zhang, Yujin Yuan, Libin Zheng, Bo Zheng, et al. Comi: Coarse-to-fine context compression via marginal information gain. In *International Conference on Learning Representations*, volume 2026, pages 46648–46666, 2026. 2, 3, 7, 15

[36] Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286, 2012. 2, 4, 17

[37] Shaoshen Chen, Yangning Li, Zishan Xu, Yongqin Zeng, Shunlong Wu, Xinshuo Hu, Zifei Shan, Xin Su, Jiwei Tang, Yinghui Li, et al. Dast: Context-aware compression in llms via dynamic allocation of soft tokens. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20544–20552, 2025. 3

[38] Ling Xing, Alex Jinpeng Wang, Rui Yan, Xiangbo Shu, and Jinhui Tang. Vision-centric token compression in large language model. In *Advances in Neural Information Processing Systems*, volume 38, pages 33080–33110, 2025. 3

[39] 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 (Volume 1: Long Papers)*, pages 25081–25091, 2025. 3

[40] Jiwei Tang, Zhicheng Zhang, Shunlong Wu, Jingheng Ye, Lichen Bai, Zitai Wang, Tingwei Lu, Lin Hai, Yiming Zhao, Hai-Tao Zheng, et al. Gmsa: Enhancing context compression via group merging and layer semantic alignment. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28690–28704, 2026. 3

[41] Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, et al. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. *arXiv preprint arXiv:2403.12968*, 2024. 3, 7

[42] Yi Zhao, Zuchao Li, Hai Zhao, Baoyuan Qi, and Liu Guoming. Dac: A dynamic attention-aware approach for task-agnostic prompt compression. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19395–19407, 2025. 3

[43] 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. *arXiv preprint arXiv:2310.06839*, 2023. 3, 7

[44] Jiwei Tang, Jin Xu, Tingwei Lu, Zhicheng Zhang, Yiming Zhao, Lin Hai, and Hai-Tao Zheng. Perception compressor: A training-free prompt compression framework in long context scenarios. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4093–4108, 2025. 3

[45] Yunlong Zhao, Haoran Wu, and Bo Xu. Leveraging attention to effectively compress prompts for long-context llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26048–26056, 2025. 3

[46] Yihang Wang, Xu Huang, Bowen Tian, Yueyang Su, Lei Yu, Huaming Liao, Yixing Fan, Jiafeng Guo, and Xueqi Cheng. Quito-x: A new perspective on context compression from the information bottleneck theory. *arXiv preprint arXiv:2408.10497*, 2024. 3

[47] Quancai Liu, Haihui Fan, Jinchao Zhang, Xiangfang Li, Chuanrong Li, and Bo Li. Discomp: A two-stage prompt optimization framework combining task-agnostic and task-aware compression. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1033–1044, 2025. 3

[48] Shivam Shandilya, Menglin Xia, Supriyo Ghosh, Huiqiang Jiang, Jue Zhang, Qianhui Wu, Victor Rühle, and Saravan Rajmohan. Taco-rl: Task aware prompt compression optimization with reinforcement learning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1582–1597, 2025. 3

[49] JungMin Yun and Youngbin Kim. Itercomp: Reasoning-aware adaptive prompt compression for multi-hop question answering. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33827–33840, 2026. 3

[50] Shenglai Zeng, Tianqi Zheng, Chuan Tian, Dante Everaert, Yau-Shian Wang, Yupin Huang, Michael J Morais, Rohit Patki, Jinjin Tian, Xinnan Dai, et al. Attn-gs: Attention-guided context compression for efficient personalized llms. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 35925–35946, 2026. 3

[51] Hongjin Qian, Zheng Liu, Peitian Zhang, Zhicheng Dou, and Defu Lian. Boosting long-context information seeking via query-guided activation refilling. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9453–9464, 2025. 3

[52] Zhuoen Chen, Dongfang Li, Meishan Zhang, Baotian Hu, and Min Zhang. Dynamic long context reasoning over compressed memory via end-to-end reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8064–8083, 2026. 3

[53] Sumin An, Junyoung Sung, Wonpyo Park, Chanjun Park, and Hongsuck Seo. Lcirc: A recurrent compression approach for efficient long-form context and query dependent modeling in llms. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10431–10442, 2025. 3

[54] Eunseong Choi, Sunkyung Lee, Minjin Choi, Jun Park, and Jongwuk Lee. From reading to compressing: Exploring the multi-document reader for prompt compression. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14734–14754, 2024. 3

[55] Yang Xu, Yunlong Feng, Honglin Mu, Yutai Hou, Yitong Li, Xinghao Wang, Wanjun Zhong, Zhongyang Li, Dandan Tu, Qingfu Zhu, et al. Concise and precise context compression for tool-using language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16430–16441, 2024. 3

[56] 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 *EMNLP*, pages 2369–2380, 2018. 7, 15

[57] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, 2020. 7, 9, 15

[58] 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*, 7:453–466, 2019. 7, 8, 15

[59] Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018. 7, 15

[60] Alexander Richard Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1074–1084, 2019. 7, 15

[61] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3119–3137, 2024. 7

[62] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 7, 8

[63] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 7

[64] Zhiwei Cao, Qian Cao, Yu Lu, Ningxin Peng, Luyang Huang, Shanbo Cheng, and Jinsong Su. Retaining key information under high compression ratios: Query-guided compressor for llms. In *ACL (1)*, pages 12685–12695. Association for Computational Linguistics, 2024. 15

# APPENDIX

The document is organized as follows:

- §A Problem Formulation.
- §B Experiment Details.
- §C More Ablation Studies.
- §D Set-Conditioned Allocation Overhead.
- §E Determinantal Point Processes for Set-Conditioned Context Compression.

## A PROBLEM FORMULATION

Given a query sequence $x_q$ and a long context divided into $N$ segments $\{x_i\}_{i=1}^{N}$, our goal is to construct a query-aware compressed context under a target compression ratio $\rho$, while preserving information essential for downstream inference. We adopt a hybrid representation in which a subset of context segments retains its full token-level representations, while the remaining segments are represented compactly. Let $\mathcal{Y} \subseteq \{1, \ldots, N\}$ denote the indices of the high-fidelity segments, with $|\mathcal{Y}| = K$. Given a segment length $L_{\text{seg}}$, the high-fidelity budget is determined by $K = \left\lfloor \frac{NL_{\text{seg}}/\rho - N}{L_{\text{seg}} - 1} \right\rfloor$. For each segment $x_i$, its representation fidelity is determined by whether it belongs to $\mathcal{Y}$. Segments in $\mathcal{Y}$ retain their original token-level representations $\mathbf{H}_i$, while the remaining segments are represented by their compact representations $\mathbf{c}_i$. The resulting segment representations $\mathbf{R}_i$ are restored to their original context order and concatenated to form the hybrid compressed context $\mathbf{R}$ for downstream generation. The central challenge is to determine the high-fidelity set $\mathcal{Y}$ under the limited budget, considering both the relevance of individual segments to the query and their marginal contributions relative to the information already preserved at full fidelity.

## B EXPERIMENT DETAILS

### B.1 BASELINE IMPLEMENTATION

RAM [34], COMI [35], and ICAE [28] are trained under the same experimental settings as `ReValue` for a fair comparison. Results for the remaining baselines are taken from RAM [34]. Following the evaluation setup of [34], we evaluate our method on five datasets: HotpotQA [56], 2WikiMultiHopQA (2WikiMQA) [57], Natural Questions (NaturalQA) [58], NarrativeQA (NaQA) [59], and Multi-News (News) [60]. The Original Prompt results correspond to zero-shot inference with the uncompressed context, without additional task-specific fine-tuning. During training, both the compressor encoder and the downstream LLM are trainable.

### B.2 DATASET DETAILS

**NaturalQuestions** This dataset consists of real queries issued to the Google search engine, paired with entire Wikipedia pages. It requires models to identify both a long answer (typically a paragraph) and a short answer (one or more entities or a boolean) within the document. The corpus contains 307,373 training examples, 7,830 development examples, and 7,842 test examples. It is widely utilized for evaluating open-domain question answering and machine reading comprehension under realistic information-seeking scenarios. *We select the processed version [64] where each question has 20 related documents and only one of them contains the correct answer. The processed version has 2,655 test samples.* We report best-subspan Exact Match (EM) and token-level F1.

**HotpotQA** HotpotQA is a large-scale dataset designed for multi-hop reasoning over Wikipedia articles. Questions are constructed such that the answer can only be found by performing reasoning across multiple documents. It features 112,779 training samples and 7,405 validation samples. A distinguishing factor of this dataset is the requirement for models to provide supporting facts (sentences) to explain the reasoning process, which enhances the explainability of the question-answering systems. *We use the validation set to evaluate model's performance.*

**2WikiMQA** This dataset focuses on multi-hop question answering with structured evidence, specifically utilizing Wikipedia and Wikidata. It aims to minimize the presence of "reasoning shortcuts" by using templates to generate complex questions that require synthesizing information from up to four documents. The dataset includes 167,454 training instances, 12,576 validation instances, and 12,576 test instances. It includes four types of reasoning: compositional, inference, comparison, and bridge. *We use the test set to evaluate model performance.*

**NarrativeQA** NarrativeQA is designed to test deep understanding of entire stories rather than local surface-level matching. It contains questions based on complete books and movie scripts. Unlike datasets that rely on short snippets, this benchmark requires models to capture long-range dependencies. The collection includes 1,572 documents divided into 1,102 for training, 115 for validation, and 355 for testing. These documents correspond to 46,765 total question and answer pairs. *We use the test set to evaluate the model's performance, filtering out test samples longer than 32K.*

**MultiNews** As a large-scale multi-document summarization dataset, Multi-News consists of news articles and human-written summaries. Each sample contains a summary paired with multiple source articles from various news sites. The dataset provides 44,972 training examples, 5,622 validation examples, and 5,622 test examples. It is a standard benchmark for evaluating the ability of models to consolidate redundant information and resolve conflicting details across diverse sources. *We use the test set to evaluate model performance.*

## C More Ablation Studies

**Effect of Stop-Gradient in Regularization.** We examine the effect of detaching the quality weights in $\mathcal{L}_{\text{REG}}$ on NaturalQA under $2\times$, $4\times$, and $8\times$ compression. As shown in Table 7, stop-gradient consistently improves EM/F1 by 2.78/2.36, 8.82/7.28, and 10.40/8.76 points, respectively, with larger gains under higher compression ratios. By preventing regularization gradients from propagating through the quality-weight branch while preserving their effect on compact segment representations through the similarity matrix, stop-gradient separates direct optimization of the quality weights from representation diversity regularization. These results demonstrate the benefit of this gradient separation, particularly under tighter representation budgets.

Table 7: **Effect of stop-gradient in the regularization objective.** All other components are kept unchanged. The best results under each compression ratio are shown in **bold**.

| Setting | $2\times$ | | $4\times$ | | $8\times$ | |
|---|---|---|---|---|---|---|
| | **EM** | **F1** | **EM** | **F1** | **EM** | **F1** |
| w/o Stop-Gradient | 64.07 | 58.74 | 52.69 | 49.28 | 49.00 | 45.84 |
| w/ Stop-Gradient | **66.85** | **61.10** | **61.51** | **56.56** | **59.40** | **54.60** |

**Effect of Segment Size.** Table 8 studies the effect of segment granularity under $4\times$ compression. HotpotQA remains largely stable when the segment size increases from 50 to 100, whereas 2WikiMQA consistently benefits from finer segmentation. Further increasing the segment size to 200 causes a clear performance drop on both datasets. Since our framework allocates representation fidelity at the segment level, overly large segments make this allocation coarser by grouping more relevant and less relevant content into the same decision unit. These results suggest that finer segmentation provides a more suitable granularity for fidelity allocation in our framework. We therefore adopt a segment size of 50 as the default setting.

## D Set-Conditioned Allocation Overhead

We measure the latency of set-conditioned high-fidelity allocation on Qwen3-4B-Instruct across input context lengths from 4K to 64K tokens, including both multi-scale kernel construction and greedy selection. As shown in Table 9, the allocation latency increases from 4.74 ms at 4K tokens to 57.40

Table 8: **Effect of the segment size.**

| Segment Size | HotpotQA | | 2WikiMQA | | AVG | |
|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 |
| 50 | 47.49 | **60.07** | **50.26** | **56.10** | **48.88** | **58.09** |
| 100 | **47.77** | 59.89 | 47.21 | 52.79 | 47.49 | 56.34 |
| 200 | 41.85 | 53.76 | 42.42 | 47.48 | 42.14 | 50.62 |

ms at 64K tokens. Even at the longest evaluated context length, the allocation process completes within 60 ms, indicating that set-conditioned allocation introduces limited additional latency.

Table 9: **Latency of set-conditioned allocation.** Allocation latency includes multi-scale kernel construction and greedy high-fidelity selection.

| Input Length | 4K | 8K | 16K | 32K | 64K |
|---|---|---|---|---|---|
| Allocation Latency (ms) | 4.74 | 6.46 | 10.39 | 27.29 | 57.40 |

# E Determinantal Point Processes for Set-Conditioned Context Compression

**DPP Preliminaries.** Determinantal Point Processes (DPPs) [36] define a probability distribution over subsets of a finite ground set, assigning higher probabilities to subsets whose elements have high individual quality and low mutual similarity. Given a positive semidefinite kernel $\mathbf{L} \in \mathbb{R}^{N\times N}$, an $L$-ensemble DPP assigns the following probability to a subset $\mathcal{Y} \subseteq \{1, \ldots, N\}$:

$$P(\mathcal{Y}) = \frac{\det(\mathbf{L}_{\mathcal{Y}})}{\det(\mathbf{I}_N + \mathbf{L})}, \tag{12}$$

where $\mathbf{L}_{\mathcal{Y}}$ denotes the principal submatrix of $\mathbf{L}$ indexed by $\mathcal{Y}$.

In our context compression setting, each element corresponds to a context segment, and the objective is to select exactly $K$ segments for high-fidelity preservation. Conditioning the DPP on the subset cardinality yields a $K$-DPP, which assigns the following probability to each subset of size $K$:

$$P(\mathcal{Y} \mid |\mathcal{Y}| = K) = \frac{\det(\mathbf{L}_{\mathcal{Y}})}{\sum_{\substack{\mathcal{Z}\subseteq\{1,\ldots,N\} \\ |\mathcal{Z}|=K}} \det(\mathbf{L}_{\mathcal{Z}})}, \tag{13}$$

provided that the normalization term is positive.

Since the normalization term is independent of the selected subset, maximum a posteriori (MAP) selection under the cardinality constraint is equivalent to maximizing $\log\det(\mathbf{L}_{\mathcal{Y}})$. This gives the set objective in Eq. 5, which `ReValue` approximately solves through greedy selection rather than sampling from the DPP distribution.

**Connection to Set-Conditioned Valuation.** A key property of the DPP formulation is that the value of adding a candidate depends on the elements already included in the selected set. Consider a current high-fidelity set $\mathcal{Y}$ and a candidate segment $i \notin \mathcal{Y}$. When $\mathbf{S}_{\mathcal{Y}}$ is nonsingular, the determinant of the expanded similarity matrix can be decomposed using the Schur complement:

$$\begin{aligned} \det(\mathbf{S}_{\mathcal{Y}\cup\{i\}}) &= \det(\mathbf{S}_{\mathcal{Y}}) \left(S_{ii} - \mathbf{S}_{i\mathcal{Y}}\mathbf{S}_{\mathcal{Y}}^{-1}\mathbf{S}_{\mathcal{Y}i}\right) \\ &= \det(\mathbf{S}_{\mathcal{Y}}) \left(1 - \mathbf{S}_{i\mathcal{Y}}\mathbf{S}_{\mathcal{Y}}^{-1}\mathbf{S}_{\mathcal{Y}i}\right), \end{aligned} \tag{14}$$

where the second equality follows from $S_{ii} = 1$.

Combining this decomposition with the quality–diversity factorization in Eq. 6 gives the marginal gain

$$\Delta(i \mid \mathcal{Y}) = 2\log a_i + \log\left(1 - \mathbf{S}_{i\mathcal{Y}}\mathbf{S}_{\mathcal{Y}}^{-1}\mathbf{S}_{\mathcal{Y}i}\right), \tag{15}$$

which is equivalent to Eq. 7 in the main text.

The first term measures the query-conditioned quality of candidate $i$, while the second depends on its similarity to the current high-fidelity set and the similarity structure within that set. Unlike a fixed segment-wise score, the marginal gain therefore changes as the selected set evolves.

To further interpret the set-dependent term, consider the feature-space representation of the positive semidefinite similarity kernel. Let $\boldsymbol{\phi}_i$ denote the corresponding feature representation of segment $i$, such that $S_{ij} = \langle \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \rangle$. Since $S_{ii} = 1$, each feature representation has unit norm.

Let $\mathcal{P}_{\mathcal{Y}}$ denote the orthogonal projection onto the span of the feature representations associated with the selected segments. For a nonsingular $\mathbf{S}_{\mathcal{Y}}$, the Schur complement satisfies

$$\begin{aligned} 1 - \mathbf{S}_{i\mathcal{Y}} \mathbf{S}_{\mathcal{Y}}^{-1} \mathbf{S}_{\mathcal{Y}i} \\ = \left\| \left( \mathbf{I} - \mathcal{P}_{\mathcal{Y}} \right) \boldsymbol{\phi}_i \right\|^2 . \end{aligned} \tag{16}$$

This identity shows that the set-dependent term measures the squared residual of the candidate's kernel feature representation after projection onto the subspace spanned by the selected segments. A candidate whose feature representation is largely captured by this subspace receives a stronger redundancy penalty, whereas a candidate with a larger residual retains a higher marginal value.

Moreover, as the high-fidelity set expands, the subspace spanned by its feature representations can only grow. Consequently, the residual norm of a fixed candidate cannot increase, and its set-dependent redundancy penalty cannot become less severe. This provides a mathematical interpretation of how `ReValue` continually re-evaluates each candidate relative to the evolving high-fidelity set.

**Validity of the Multi-Scale DPP Kernel.** A valid $L$-ensemble DPP requires its kernel to be positive semidefinite. In our formulation, each Gaussian similarity matrix $\boldsymbol{\kappa}^{(m)}$, defined in Eq. 3, is positive semidefinite because the Gaussian radial basis function is a positive semidefinite kernel for any bandwidth $\sigma_m > 0$.

The multi-scale similarity matrix is constructed as

$$\mathbf{S} = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{\kappa}^{(m)}. \tag{17}$$

Since a nonnegative weighted sum of positive semidefinite matrices remains positive semidefinite, $\mathbf{S}$ is also positive semidefinite.

Furthermore, the DPP kernel is constructed as $\mathbf{L} = \mathbf{ASA}$, where $\mathbf{A} = \mathrm{Diag}(a_1, \ldots, a_N)$ contains the query-conditioned quality weights. For any vector $\mathbf{v} \in \mathbb{R}^N$, we have

$$\mathbf{v}^\top \mathbf{L} \mathbf{v} = (\mathbf{A}\mathbf{v})^\top \mathbf{S} (\mathbf{A}\mathbf{v}) \geq 0. \tag{18}$$

Therefore, $\mathbf{L}$ is positive semidefinite and defines a valid DPP kernel.

This construction combines query-conditioned segment quality and multi-scale inter-segment similarity within a unified set objective. The quality weights characterize individual segment relevance, while the similarity matrix determines how each candidate's marginal contribution depends on the current high-fidelity set.

**Interpretation of the Diversity Regularizer.** The DPP formulation also motivates the representation diversity regularizer introduced in Sec. 3.4. For an $L$-ensemble DPP, the normalization term admits the following expansion:

$$\det(\mathbf{I}_N + \widetilde{\mathbf{L}}) = \sum_{\mathcal{Y} \subseteq \{1, \ldots, N\}} \det(\widetilde{\mathbf{L}}_{\mathcal{Y}}), \tag{19}$$

where the determinant of the empty principal submatrix is defined as 1. This identity shows that the normalization term aggregates the determinants of all possible subsets, rather than only those satisfying the high-fidelity budget. Maximizing its logarithm encourages compact segment representations to support subsets with larger joint determinants.

To illustrate the effect of the regularizer on pairwise similarity, consider two segments $i$ and $j$ with fixed quality weights. The determinant of their regularized kernel is

$$\begin{aligned}&\det\left(\mathbf{I}_2+\widetilde{\mathbf{L}}_{\{i,j\}}\right)\\&\quad=(1+a_i^2)(1+a_j^2)-a_i^2a_j^2S_{ij}^2.\end{aligned}\tag{20}$$

For fixed quality weights, a lower pairwise similarity $S_{ij}$ increases this determinant, illustrating how the regularizer discourages excessive similarity among compact segment representations. Since the quality weights are detached, the regularization loss propagates gradients through the multi-scale similarity kernel rather than through the quality-weight factors. This encourages more discriminative compact representations for constructing the set-conditioned allocation kernel.