LEARNING CONTEXT COMPRESSION FOR LLMS VIA IMPLICIT SEMANTIC–STRUCTURAL DECOMPOSITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Context compression enables large language models (LLMs) to reuse long contexts through compact continuous representations, yet existing learning objectives such as reconstruction, continuation prediction, and question answering largely supervise each context in isolation. We revisit context compression from a new semantic–structural perspective, where an effective compressed representation should remain stable under structural changes that preserve meaning, while remaining sensitive to semantic changes under largely preserved structure. Realizing this principle is nontrivial because semantics and linguistic structure are intrinsically entangled and cannot be explicitly separated at the token or representation level. We introduce Semantic–Structural Context Compression (S^2C), which realizes this decomposition implicitly through paired context variants and learning objectives that remain fully compatible with the LLM's native next-token prediction framework. Shared-Target Generation (STG) learns structural invariance by requiring structurally different but semantically equivalent contexts to generate a shared third-view target, thereby suppressing source-specific linguistic form from direct supervision. Response-Shift Matching (RSM) complements this by preserving semantic sensitivity, matching the prediction shifts induced by localized semantic edits under the full context rather than supervising the edited examples independently. Together, STG and RSM separate the effects of structural and semantic variation without explicit latent-space disentanglement or modification of the underlying compression architecture. S^2C consistently outperforms existing context compression methods on six benchmarks, with particularly strong gains at high compression ratios.

1 INTRODUCTION

As large language models (LLMs) expand toward longer-context understanding and complex reasoning, context compression (Ge et al., 2023; Li et al., 2025; Liu et al., 2026; Jiang et al., 2023; 2024; Mu et al., 2023; Chevalier et al., 2023; Zhang et al., 2025; Tang et al., 2026a; Zhao et al., 2024; Tang et al., 2026b;c) has become increasingly important for reducing the computational and memory costs of long-context processing. Unlike retrieval, token pruning, or prompt shortening (Li et al., 2023; Anagnostidis et al., 2023; Xu et al., 2024; Chen et al., 2024; Cao et al., 2024; Cheng et al., 2024), which reduce the input itself, soft context compression encodes the full context into compact continuous representations that can be reused across downstream queries for generation, question answering, and reasoning. Existing work has explored various compression architectures and representation forms, using objectives such as reconstruction, continuation prediction, and question answering (Ge et al., 2023; Qin et al., 2024; Liu & Qiu, 2025). Yet these methods largely treat each context as an independent training instance: they specify what should be produced from a given context, but provide little explicit guidance on **how different types of information in the context should be jointly preserved under compression**.

We revisit context compression from a new perspective. Rather than treating text as a monolithic token sequence, we conceptually distinguish two intertwined factors: **semantic content**, such as entities, attributes, facts, relations, and values, and the **linguistic structure** through which that content is expressed, including lexical choice, syntax, and discourse organization. Importantly, our goal is not to prioritize semantics while discarding structure. Both are essential to language understanding and generation, but they play different roles under compression. Substantial changes in linguistic

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.

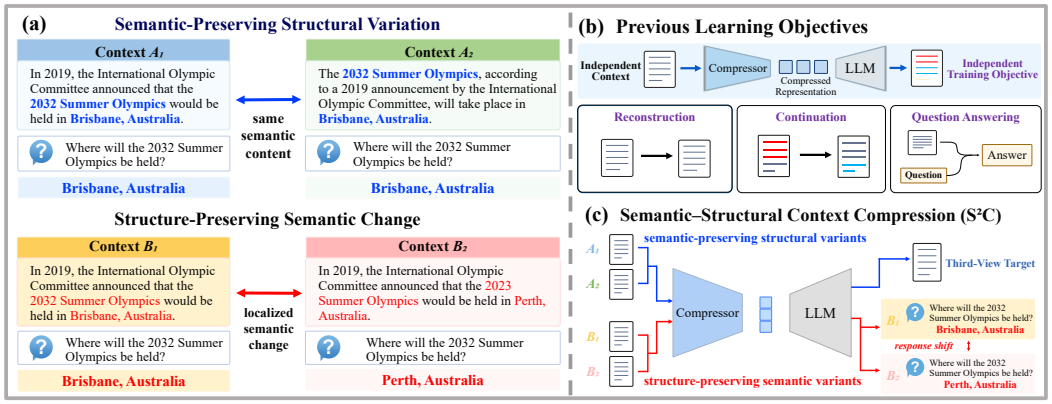


Figure 1: **From isolated-context learning to semantic–structural context compression.** (a) Two complementary context variations we identified: semantic-preserving structural variation and structure-preserving semantic change. (b) Existing methods use reconstruction, continuation prediction, or question answering objectives, but still treat each context as an independent training instance. (c) Our S^{2}C learns across related contexts, using shared third-view supervision to reduce dependence on source-specific linguistic structure and response-shift matching to preserve the effects of semantic changes, all within the LLM's native next-token prediction framework.

structure may preserve the same semantic content, while a localized semantic change under largely preserved structure may alter a critical fact and its downstream predictions, as shown in Figure 1. Effective compression should learn both sides of this relationship: **invariance to semantic-preserving structural variation and sensitivity to structure-preserving semantic change**.

The key difficulty is that semantics and linguistic structure are intrinsically entangled. Linguistic form itself carries meaning, while semantic content is necessarily realized through linguistic form. They therefore cannot be cleanly separated at either the token or representation level, making explicit semantic–structural disentanglement both difficult and potentially undesirable. Our key idea is to realize this distinction **implicitly through the learning objective**. We construct two complementary context variants by varying one factor while keeping the other approximately fixed: a *semantic-preserving structural variant*, which changes the linguistic realization while preserving meaning, and a *structure-preserving semantic variant*, which largely preserves linguistic form while introducing a localized semantic edit. These variants make it possible to study structural and semantic variation separately without explicitly decomposing the representation itself.

A natural way to exploit these variants would be to directly constrain the compressed representations, for example by pulling semantically equivalent contexts closer and pushing semantically different ones apart. However, semantic relations do not imply a predefined geometry in the compression space, and directly enforcing such alignment may interfere with the representation structure already learned by the pretrained LLM. Moreover, geometric similarity alone does not ensure that the preserved information can actually be read and used by the decoder. We therefore seek to realize semantic–structural learning in the **LLM output space**, where supervision remains directly tied to the model's own prediction form.

This, however, introduces a second technical challenge. Simply moving supervision to the output space is insufficient because standard next-token prediction still entangles semantics and structure. If each context reconstructs its own text, different linguistic realizations produce different token-level targets even when they express the same semantics. Likewise, independently supervising an original context and its semantic edit does not capture how the model's predictions should change because of that edit. The central technical question is therefore: **how can semantic and structural variation be learned separately while remaining fully compatible with the LLM's native next-token prediction framework?**

To address this challenge, we introduce **Semantic–Structural Context Compression (S^{2}C)**, which uses two complementary learning objectives:

- **Shared-Target Generation (STG).** For semantic-preserving structural variants, we avoid both source-specific reconstruction and direct representation alignment. Instead, two structurally different source contexts are trained to generate the same third semantically equivalent context. By

sharing both the target sequence and its autoregressive next-token supervision, source-specific wording and syntax no longer define separate prediction targets, while information shared across different linguistic structures remains consistently required for generation. STG therefore learns structural invariance without explicitly separating structure from the compressed representation.

- **Response-Shift Matching (RSM).** For structure-preserving semantic variants, independently supervising the original and edited contexts still fails to isolate the effect of the semantic change itself. We therefore measure the response shift induced by the edit under the full context and require the compressed context to reproduce the same relative shift. RSM thus learns semantic sensitivity by explicitly modeling how predictions should change for queries affected by the edit, while keeping unrelated responses stable.

Together, STG and RSM provide a way to learn the two coupled factors **separately but implicitly**: STG suppresses source-specific structural variation through shared third-view generation, while RSM preserves semantic variation through response-shift matching. Crucially, both objectives are realized within the LLM's native autoregressive prediction framework, without explicit latent-space disentanglement, geometric constraints on the compression space, or changes to the underlying compression mechanism.

Experiments on SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2017), TriviaQA (Joshi et al., 2017), SearchQA (Dunn et al., 2017), HotpotQA (Yang et al., 2018), and Natural Questions (Kwiatkowski et al., 2019) show that S^2C consistently improves over existing context compression methods, with larger gains at higher compression ratios. Further analysis demonstrates stronger out-of-domain generalization while maintaining competitive context reconstruction performance. Our main contributions are:

- We revisit context compression from the **learning-objective perspective**, showing that effective compression should distinguish semantic-preserving structural variation from structure-preserving semantic change, rather than treating each context as an isolated training instance.
- We identify the central challenge that semantics and structure are intrinsically entangled, and realize their decomposition **implicitly through paired context variants** that vary one factor while approximately preserving the other.
- We develop two novel output-space objectives that realize this decomposition **within the LLM's native next-token prediction framework**: STG learns structural invariance through shared third-view generation, while RSM preserves semantic sensitivity by matching edit-induced response shifts measured under the full context.
- S^2C consistently outperforms existing context compression methods across six benchmarks, with larger gains at high compression ratios, stronger out-of-domain generalization, and competitive reconstruction performance, all without modifying the underlying compression architecture.

2 METHOD

The overall framework is illustrated in Figure 2. We study the learning objective for soft context compression, with the goal of jointly modeling semantic content and linguistic structure within a compact context representation. Since the two factors are intrinsically entangled, Sec. 2.1 introduces an implicit semantic–structural decomposition through complementary context variants and formulates the corresponding output-space learning principle. Sec. 2.2 then introduces Shared-Target Generation (STG) to learn invariance to semantic-preserving structural variation, while Sec. 2.3 presents Response-Shift Matching (RSM) to preserve sensitivity to structure-preserving semantic change.

2.1 SEMANTIC–STRUCTURAL MODELING FOR CONTEXT COMPRESSION

Problem setup. We focus on the learning objective for soft context compression rather than a specific architecture. Following the commonly used encoder–decoder formulation (Ge et al., 2023), a compressor $\mathcal{C}_\theta$ maps an input context $X = \{x_t\}_{t=1}^{N}$ into M compact continuous representations $Z = \mathcal{C}_\theta(X) = \{z_j\}_{j=1}^{M}$, where $M \ll N$. The compressed representation Z replaces the original context as input to the LLM decoder $\mathcal{D}_\phi$, where ϕ denotes the LoRA parameters. Given a downstream query Q, the decoder predicts the target output A conditioned on (Z, Q). Depending on the training objective, A may correspond to the original context for reconstruction, a subsequent continuation, or an answer to a question about the context. For instance, under a reconstruction objective,

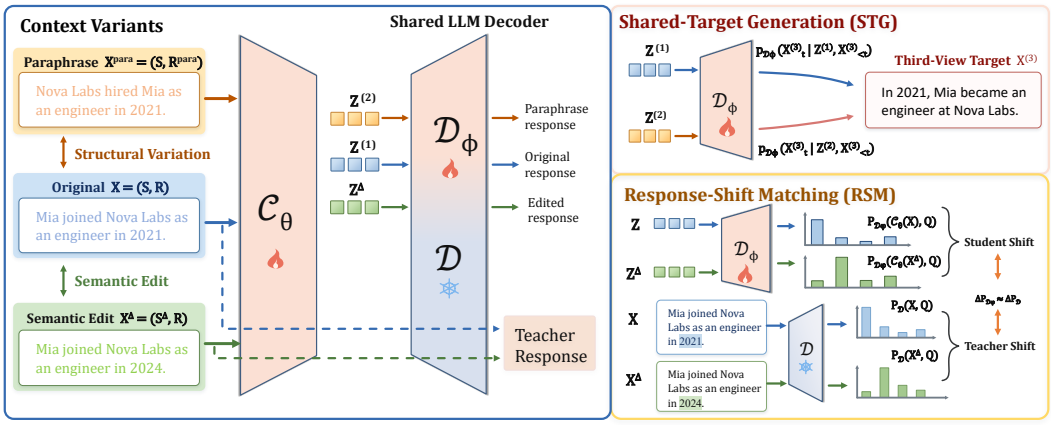


Figure 2: **Framework of S^{2}C.** A shared compressor and LLM decoder process the original context and its semantic-preserving structural and structure-preserving semantic variants. **STG** learns structural invariance by mapping semantically equivalent contexts to a shared third-view target, while **RSM** preserves semantic sensitivity by matching edit-induced response shifts between the compressed student and full-context teacher.

the decoder autoregressively recovers the original context from the compressed representation:

$$\mathcal{L}_{\text{reconstruction}} = \sum_{t=1}^{N} \log p_{\mathcal{D}\phi}\left(x_t \mid Z, x_{<t}\right), \tag{1}$$

where $x_{<t}$ denotes the ground-truth prefix preceding x_t. Despite using different prediction targets, existing objectives generally supervise each context independently, leaving how compression should respond to semantic versus structural variation unspecified.

Semantic–Structural Decomposition. Existing objectives typically treat each context as a single training instance, without explicitly distinguishing different sources of textual variation. We instead conceptually characterize a context by two intertwined factors:

$$X = (S, R), \tag{2}$$

where S denotes its semantic content, including entities, attributes, facts, relations, and values, and R denotes the linguistic structure through which that content is expressed, including lexical choice, syntax, and discourse organization. Importantly, semantics and linguistic structure are inherently coupled in natural language and cannot be explicitly separated at either the token or representation level. We therefore realize this decomposition implicitly by varying one factor while keeping the other approximately fixed. Specifically, we construct

$$X^{\text{para}} = (S, R^{\text{para}}), \qquad X^{\Delta} = (S^{\Delta}, R), \text{ where } S^{\Delta} = S \oplus \Delta S. \tag{3}$$

Here, X^{para} is a semantic-preserving structural variant that changes the linguistic realization while preserving meaning, whereas X^{Δ} is a structure-preserving semantic variant that introduces a localized semantic edit ΔS while keeping the surrounding linguistic structure as unchanged as possible.

These two variants expose complementary forms of variation: substantial changes in linguistic form may preserve the same semantics, whereas a small localized edit may alter a critical fact. They therefore provide an implicit way to isolate structural and semantic effects without explicitly decomposing the compressed representation.

Output-Space Semantic–Structural Constraints. Given X, X^{para}, and X^{Δ}, we obtain their compressed representations using the shared compressor $\mathcal{C}_\theta$. Ideally, compression should preserve two complementary properties: invariance to semantic-preserving structural variation and sensitivity to structure-preserving semantic change.

A direct approach is to enforce these properties in the compression space by pulling the representations of X and X^{para} closer while pushing that of X^{Δ} away. However, semantic relations do not imply a predefined representation geometry, and geometric proximity or separation does not guarantee that the encoded information can be correctly read and used by the LLM decoder. We therefore

impose these constraints in the LLM output space. Given a query Q, the desired behavior is

$$\begin{cases} \mathcal{D}_\phi\left(\mathcal{C}_\theta(S,R),Q\right) \simeq \mathcal{D}_\phi\left(\mathcal{C}_\theta(S,R^{\text{para}}),Q\right), & \text{for structural variation,} \\ \mathcal{D}_\phi\left(\mathcal{C}_\theta(S,R),Q\right) \neq \mathcal{D}_\phi\left(\mathcal{C}_\theta(S^{\Delta},R),Q\right), & \text{if } Q \text{ depends on } \Delta S. \end{cases} \tag{4}$$

Thus, changes in linguistic structure should leave the relevant model behavior unchanged when semantics are preserved, whereas a localized semantic edit should induce corresponding changes only when it affects the query.

Importantly, standard next-token supervision does not naturally enforce these properties: reconstructing different linguistic realizations yields different token-level targets even when their semantics are identical, while independently supervising the original and semantically edited contexts does not capture how predictions should change in response to the edit. Secs. 2.2 and 2.3 address these two challenges through Shared-Target Generation and Response-Shift Matching, respectively.

2.2 Shared-Target Generation

Motivation. For semantic-preserving structural variants, the desired property is invariance to source-specific linguistic realization. Standard reconstruction, however, assigns a different token-level target to each realization. Consider two semantically equivalent contexts

$$X^{(1)} = (S, R^{(1)}), \qquad X^{(2)} = (S, R^{(2)}). \tag{5}$$

Reconstructing $X^{(1)}$ and $X^{(2)}$ independently directly rewards their respective lexical and syntactic choices, even though these differences do not change the underlying semantics. Directly aligning their compressed representations could remove this discrepancy, but would require imposing an explicit geometry on the compression space. We instead seek to learn structural invariance through the LLM's native next-token prediction objective.

Shared-target generation. To this end, we reduce the influence of source-specific structural variation on the supervision signal by replacing source-specific reconstruction with a shared third-view generation target. Specifically, given three semantically equivalent contexts with different linguistic structures,

$$X^{(1)} = (S, R^{(1)}), \qquad X^{(2)} = (S, R^{(2)}), \qquad X^{(3)} = (S, R^{(3)}), \tag{6}$$

we compress the first two contexts as

$$Z^{(i)} = \mathcal{C}_\theta(X^{(i)}), \qquad i \in \{1, 2\}, \tag{7}$$

and use $X^{(3)}$ as their shared generation target. Crucially, the shared target is distinct from both source contexts. Using either source as the target would still reward its own linguistic structure, whereas the third-view target provides a common output realization independent of both source formulations. Under teacher forcing, the target prefix further provides both branches with the same target-side linguistic context, implicitly anchoring generation to the linguistic structure of $X^{(3)}$.

Accordingly, at each decoding step, both branches condition on the same prefix and predict the same next token:

$$p_{\mathcal{D}_\phi}\left(X_t^{(3)} \mid Z^{(1)}, X_{<t}^{(3)}\right) \simeq p_{\mathcal{D}_\phi}\left(X_t^{(3)} \mid Z^{(2)}, X_{<t}^{(3)}\right), \tag{8}$$

Importantly, STG does not require the two branches to match their full predictive distributions. Instead, they are independently optimized toward the same ground-truth next token under the same autoregressive context, enforcing the desired consistency while preserving flexibility in their remaining output distributions.

Training objective. The STG objective is implemented with the standard autoregressive likelihood:

$$\mathcal{L}_{\text{STG}} = -\sum_{i\in\{1,2\}} \sum_{t=1}^{T} \log p_{\mathcal{D}_\phi}\left(X_t^{(3)} \mid \mathcal{C}_\theta(X^{(i)}), X_{<t}^{(3)}\right), \tag{9}$$

where $T = |X^{(3)}|$. Gradients are propagated through the decoder to jointly optimize the compressor $\mathcal{C}_\theta$ and the LoRA parameters ϕ, while the pretrained LLM remains frozen. Rather than explicitly aligning $Z^{(1)}$ and $Z^{(2)}$, STG induces structural invariance through shared next-token supervision on the same third-view target, fully within the LLM's native autoregressive training framework.

2.3 RESPONSE-SHIFT MATCHING

Motivation. While STG learns invariance to semantic-preserving structural variation, compression must also remain sensitive to structure-preserving semantic changes. A localized semantic edit should alter predictions for queries that depend on the edited content, while leaving unrelated responses largely unchanged. Standard autoregressive supervision, however, treats the original context X and its edited counterpart X^{Δ} as independent training instances, and therefore does not explicitly capture how the model response should change as a consequence of the edit.

Our key insight is to characterize a semantic change by the relative response shift it induces, rather than by the two prediction endpoints independently. The full-context LLM provides a natural reference for this shift, since it directly observes the information before and after the semantic edit. We therefore require the compressed model to preserve the same response change:

$$P_{\mathcal{D}_\phi}\left(\mathcal{C}_\theta(X^{\Delta}), Q\right) - P_{\mathcal{D}_\phi}\left(\mathcal{C}_\theta(X), Q\right) \approx P_{\mathcal{D}}\left(X^{\Delta}, Q\right) - P_{\mathcal{D}}\left(X, Q\right), \tag{10}$$

where the right- and left-hand sides capture the response shifts under the full and compressed contexts, respectively. When Q depends on the edited content, the full-context shift specifies the corresponding change to be preserved; when Q is unrelated to the edit, this shift should remain small. $P_{\mathcal{D}_\phi}(\cdot, Q)$ denotes the model response to query Q, whose sequence-level score is introduced below.

Sequence-Level Response Scoring. To measure the effect of a semantic edit, we need a response score that is comparable across different contexts. Direct token-level comparison is ill-suited for this purpose, since candidate answers may differ in length and autoregressive trajectory, making their token-wise probabilities difficult to align. More importantly, the semantic consequence of an edit is reflected in the model's preference over complete answers rather than in isolated token probabilities. We therefore characterize each response at the sequence level over a shared candidate answer set.

Given $\mathcal{A} = \{A_1, \ldots, A_K\}$, where $A_k = \{a_{k,t}\}_{t=1}^{|A_k|}$ is the k-th answer, we compute a length-normalized log-likelihood for each candidate and normalize the scores into a response distribution:

$$s_k(X, Q) = \frac{1}{|A_k|} \sum_{t=1}^{|A_k|} \log p_{\mathcal{D}_\phi}\left(a_{k,t} \mid X, Q, A_{k,<t}\right), P_k(X, Q) = \frac{\exp\left(s_k(X, Q)/\tau\right)}{\sum_{j=1}^{K} \exp\left(s_j(X, Q)/\tau\right)}. \tag{11}$$

The resulting distribution $P_{\mathcal{D}_\phi}(X, Q) = \{P_k(X, Q)\}_{k=1}^{K}$ captures the model's relative preference over complete candidate answers, providing a common response space for measuring semantic-edit-induced shifts across contexts.

Teacher–Student Response Objective. We treat the frozen full-context LLM $\mathcal{D}$ as the teacher and the compressed-context model, consisting of the trainable compressor $\mathcal{C}_\theta$ and LoRA-adapted decoder $\mathcal{D}_\phi$, as the student. Freezing the teacher provides a stable reference for the response shift induced by each semantic edit.

Following Eq. 10, the core objective matches the edit-induced response shift between the full and compressed contexts. Since matching relative shifts does not constrain the absolute student response, we additionally anchor the compressed response on X to the full-context response and enforce consistency under the semantic-preserving structural variant X^{para}. The resulting objective is

$$\begin{aligned}\mathcal{L}_{\mathrm{RSM}} = {} & D_{\mathrm{KL}}\left(\mathrm{sg}[P_{\mathcal{D}}(X, Q)] \parallel P_{\mathcal{D}_\phi}\left(\mathcal{C}_\theta(X), Q\right)\right) \\ & + D_{\mathrm{JS}}\left(P_{\mathcal{D}_\phi}\left(\mathcal{C}_\theta(X), Q\right), P_{\mathcal{D}_\phi}\left(\mathcal{C}_\theta(X^{\mathrm{para}}), Q\right)\right) \\ & + \left\| \mathrm{sg}\big[P_{\mathcal{D}}(X^{\Delta}, Q) - P_{\mathcal{D}}(X, Q)\big] - \big[P_{\mathcal{D}_\phi}\left(\mathcal{C}_\theta(X^{\Delta}), Q\right) - P_{\mathcal{D}_\phi}\left(\mathcal{C}_\theta(X), Q\right)\big] \right\|_2^2,\end{aligned} \tag{12}$$

where $\mathcal{D}$ denotes the frozen full-context teacher and $\mathcal{D}_\phi \circ \mathcal{C}_\theta$ denotes the compressed-context student. The first term anchors the compressed response to the full-context response using KL divergence, with the teacher distribution serving as the fixed reference. The second enforces response consistency between X and its semantic-preserving structural variant X^{para} using the symmetric JS divergence, since the two compressed responses play equivalent roles. The third, central to RSM, matches the semantic-edit-induced response shift under compression to that observed with the full context using an ℓ_2 loss. Since response shifts are signed differences between probability distributions rather than distributions themselves, ℓ_2 directly matches both their direction and magnitude.

The stop-gradient operator $\mathrm{sg}[\cdot]$ prevents gradients from propagating through the teacher, while θ and ϕ are jointly optimized through the student branch.

Overall Learning Objective. Combining reconstruction with STG and RSM, the overall training objective is

$$\mathcal{L} = \mathcal{L}_{\text{reconstruction}} + \mathcal{L}_{\text{STG}} + \mathcal{L}_{\text{RSM}}. \tag{13}$$

Together, STG and RSM provide complementary output-space supervision for structural invariance and semantic sensitivity within the LLM's native autoregressive prediction framework.

3 Experiments

3.1 Experimental Setting

Datasets and Evaluation Metrics. Following prior studies (Liu et al., 2026), we evaluate on six question-answering benchmarks: SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2017), TriviaQA (Joshi et al., 2017), SearchQA (Dunn et al., 2017), HotpotQA (Yang et al., 2018), and Natural Questions (Kwiatkowski et al., 2019). These benchmarks cover diverse context and question types, including reading comprehension, web-search questions, and multi-hop reasoning. We report Exact Match (EM) and ROUGE-F1 (Lin, 2004) for downstream question answering, and, following SAC, additionally report perplexity (PPL) under context reconstruction to evaluate how well the compressed representation preserves information from the original context.

Baselines. We compare S^2C with three representative continuous context compression methods that isolate complementary design choices. (1) ICAE (Ge et al., 2023) is our direct baseline, using the same encoder–decoder compression architecture and differing primarily in the learning objective. (2) 500xCompressor (Li et al., 2025) represents compression-space redesign, replacing embedding-level representations with layer-wise KV states. (3) SAC (Liu et al., 2026) represents explicit semantic-aware compression, selecting semantic anchor tokens and using bidirectional attention to aggregate contextual information. We additionally report **Full-Context** and **Full-FT** as uncompressed references using the pretrained and task-fine-tuned models, respectively.

Implementation Details. We implement S^2C on the same encoder–decoder architecture as ICAE, modifying only the learning objectives while keeping the compression mechanism unchanged. Given an input context of N tokens, the compressor maps it to M continuous representations, with the compression ratio defined as N/M. Unless otherwise specified, we evaluate at $4\times$ and $16\times$ compression ratios using LLaMA-3.2-1B-Instruct (Grattafiori et al., 2024) and Qwen3-4B-Instruct (Yang et al., 2025). The compressor and decoder are initialized from the same pretrained LLM. We fully optimize the compressor parameters θ, while freezing the decoder backbone and training only its LoRA (Hu et al., 2021) parameters ϕ, with rank 128 and scaling factor 256. For fair comparison, all methods use the same training data, language-model backbone, and matched compression ratios.

Context Variants. For fair comparison, we construct both context variants from the original training corpora, without introducing additional external training sources. For *semantic-preserving structural variants*, following ICAE, we use PwC as the source corpus and prompt the base model to rewrite each document while preserving the meaning of every sentence but changing its linguistic realization. For *structure-preserving semantic variants*, following SAC, we use the six QA datasets above and provide the context, question, and gold answer to the base model, asking it to replace the answer-critical content and return the corresponding updated answer while keeping the surrounding linguistic structure largely unchanged. Prompts are provided in Appendix.

3.2 Comparison with State-of-the-Art Methods

Table 1 compares S^2C with representative context compression methods across six QA benchmarks, multiple compression ratios, and two LLM backbones. S^2C consistently achieves the strongest overall performance across these settings, with particularly large gains under more aggressive compression. Importantly, these improvements are obtained without changing the underlying compression architecture, compression space, or attention mechanism, allowing us to directly examine the benefit of the proposed semantic–structural learning objectives.

Table 1: Comparison with context compression methods. S^2C consistently outperforms all baselines across datasets, compression ratios, and LLM backbones. The gains over ICAE under the same encoder–decoder architecture isolate the benefit of our learning objectives, while improvements over 500xCompressor and SAC show that implicit semantic–structural learning remains effective without redesigning the compression space or introducing explicit semantic anchors.

Methods	SQuAD		NewsQA		TriviaQA		SearchQA		HotpotQA		NQ		Average	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Full-FT	59.71	77.69	46.04	63.50	60.54	68.80	62.07	73.25	59.26	74.78	53.47	71.01	56.85	71.51
Full-Context	16.59	38.76	9.97	26.99	35.50	49.31	5.83	12.39	22.51	39.24	18.14	38.90	18.09	34.27
LLaMA-3.2-1B-Instruct														
4× compression constraint														
ICAE (Ge et al., 2023)	26.47	42.82	17.14	29.23	52.29	61.74	55.51	64.96	35.38	49.67	37.21	55.85	37.33	50.71
500x (Li et al., 2025)	34.69	52.38	24.93	40.54	45.33	54.87	52.81	61.56	42.60	56.71	41.26	58.32	40.27	54.06
SAC (Liu et al., 2026)	44.96	67.34	28.70	49.83	56.26	68.89	58.43	70.57	50.91	67.18	45.65	66.80	47.49	65.10
Ours	**52.07**	**69.42**	**29.18**	**51.83**	**61.05**	**70.61**	**70.28**	**77.94**	**51.36**	67.12	**46.68**	**69.53**	**51.77**↑14.44	**67.74** ↑17.03
16× compression constraint														
ICAE (Ge et al., 2023)	17.04	31.17	11.30	24.35	37.48	49.52	48.63	64.71	30.86	46.90	29.23	44.58	29.09	43.54
500x (Li et al., 2025)	21.38	38.05	15.81	32.93	41.62	49.75	50.46	61.59	29.27	45.44	31.08	44.14	31.60	45.32
SAC (Liu et al., 2026)	27.57	46.61	16.38	36.44	50.73	59.82	53.06	68.30	41.18	57.26	35.97	55.11	37.48	53.92
Ours	**33.46**	**51.17**	**20.87**	**40.11**	**53.06**	**65.55**	**59.43**	**72.32**	**43.46**	**62.01**	**39.35**	**59.84**	**41.61**↑12.52	**58.50**↑14.96
Qwen3-4B-Instruct														
16× compression constraint														
ICAE (Ge et al., 2023)	26.80	43.41	14.13	28.06	47.78	55.99	56.96	72.06	34.03	51.13	34.88	51.54	35.76	50.37
500x (Li et al., 2025)	24.51	40.72	15.98	31.84	53.15	61.09	58.53	73.06	35.21	53.64	35.68	52.80	37.18	52.19
SAC (Liu et al., 2026)	27.31	45.76	19.42	35.57	55.18	67.71	59.36	71.22	38.91	55.64	37.72	55.41	39.65	55.22
Ours	**35.42**	**51.73**	**23.17**	**39.64**	**57.32**	**70.26**	**66.85**	**74.12**	**39.57**	**56.48**	**38.24**	**56.91**	**43.43**↑7.67	**58.19**↑7.82

Benefit of Learning Objectives. ICAE provides the most direct comparison because S^2C adopts the same encoder–decoder compression architecture and differs only in the learning objectives. With LLaMA-3.2-1B-Instruct, S^2C improves average EM/F1 from 37.33/50.71 to 51.77/67.74 at $4\times$ compression, corresponding to gains of 14.44 and 17.03 points. The advantage remains substantial at $16\times$, with improvements of 12.52 EM and 14.96 F1. Since the compression mechanism is unchanged, these gains directly demonstrate that how compressed representations are learned is critical beyond the architecture itself.

Comparison with Alternative Compression Designs. S^2C also outperforms methods that introduce more specialized compression mechanisms. At $16\times$ compression with LLaMA-3.2-1B-Instruct, it exceeds 500xCompressor by 10.01 EM and 13.18 F1 points, despite 500xCompressor redesigning the compression space from embedding-level representations to layer-wise KV states. The comparison with SAC is particularly informative. SAC also emphasizes semantic information, but does so explicitly by selecting semantic anchor tokens and modifying the compressor with bidirectional attention so that these anchors aggregate information from the full context. In contrast, S^2C introduces neither explicit semantic token selection nor architectural modifications, yet improves over SAC by 4.13 EM and 4.58 F1 points under the same setting.

More fundamentally, ICAE, 500xCompressor, and SAC primarily optimize or construct compressed representations from each context individually. S^2C instead explicitly learns from relations across context variants: semantic-preserving structural variation provides supervision for structural invariance, while structure-preserving semantic change provides supervision for semantic sensitivity. The consistent improvements over all three baselines therefore suggest that modeling these cross-context relations provides a complementary and effective direction.

Robustness to Compression and Model Backbone. The gains remain strong as the representation bottleneck becomes more restrictive. With LLaMA-3.2-1B-Instruct, S^2C retains improvements of 12.52 EM and 14.96 F1 over ICAE at $16\times$ compression, indicating that semantic–structural supervision becomes particularly beneficial when the available compres-

Table 2: Comparsion on out-of-domain generalization and context reconstruction.

Methods	BioASQ		DROP		RE		PPL ↓
	EM	F1	EM	F1	EM	F1	
ICAE	24.47	35.51	21.96	30.39	40.33	55.24	12.29
500x	25.93	36.30	23.55	33.46	41.11	54.37	12.17
SAC	28.66	41.31	27.48	36.72	47.90	61.04	11.46
S^2C	**30.19**	**44.05**	**28.41**	**36.94**	**51.26**	**66.71**	**11.38**

sion capacity is limited. The same trend transfers to Qwen3-4B-Instruct: at $16\times$, S^2C reaches 43.43 EM and 58.19 F1, outperforming ICAE by 7.67/7.82 points, 500xCompressor by 6.25/6.00 points, and SAC by 3.78/2.97 points. These results show that the benefit of S^2C is consistent across different model families and scales.

OOD Generalization and Reconstruction. Table 2 further evaluates generalization to unseen domains and the preservation of original context information. S^2C consistently outperforms all compression baselines on BioASQ, DROP, and RE. The improvement is particularly pronounced on RE, where S^2C improves over SAC from 47.90/61.04 to 51.26/66.71 in EM/F1. These results suggest that learning semantic-preserving structural variation and structure-preserving semantic change yields compressed representations that transfer more effectively beyond the training distribution. Importantly, this improved generalization does not come at the cost of reconstruction. S^2C achieves an LM-PPL of 11.38, comparable to and slightly better than SAC at 11.46. Despite reducing direct supervision on source-specific linguistic realizations, S^2C therefore retains the information required for context reconstruction while improving downstream generalization.

3.3 Ablation Study

We conduct ablations with Qwen3-4B-Instruct at a $16\times$ compression ratio to evaluate both the contribution of STG and RSM and the key mechanisms behind them. Results are reported in Table 3 on both ID and OOD benchmarks.

Table 3: Ablation study of different learning objectives on ID and OOD evaluation.

Method	ID		BioASQ		DROP		RE	
	Avg. EM	Avg. F1	EM	F1	EM	F1	EM	F1
S^2C	**43.43**	**58.19**	**30.19**	**44.05**	**28.41**	**36.94**	**51.26**	**66.71**
w/o $\mathcal{L}_{\text{STG}}$	38.17	53.42	29.59	43.82	27.88	36.51	50.71	66.19
w/o $\mathcal{L}_{\text{RSM}}$	40.68	54.75	21.54	33.19	20.29	28.07	44.91	60.38
$\mathcal{L}_{\text{RSM}}$ w/o Shift	41.42	54.87	26.46	41.61	22.75	31.25	47.18	62.94
STG $\rightarrow$ Source Target	34.97	47.06	24.20	39.47	25.08	33.72	42.37	56.82

Complementary roles of STG and RSM. Removing either objective consistently degrades performance. Without STG, average ID EM/F1 drops from 43.43/58.19 to 38.17/53.42, while removing RSM yields 40.68/54.75 but causes much larger OOD degradation, e.g., BioASQ drops from 30.19/44.05 to 21.54/33.19. This suggests that STG contributes strongly to overall compression quality, whereas RSM is particularly important for preserving semantic sensitivity that transfers to unseen domains.

Why a shared third-view target? Replacing STG with source-specific reconstruction reduces ID EM/F1 to 34.97/47.06, even below the variant without STG (38.17/53.42), with the same trend on OOD datasets. Thus, the gain does not simply come from additional semantically equivalent contexts. It specifically relies on the shared third-view target, which avoids directly rewarding source-specific wording and syntax.

Why match response shifts? Removing only the response-shift term while retaining response anchoring and structural consistency improves over removing RSM entirely, but remains consistently below the full model. Restoring shift matching yields especially large OOD gains, e.g., from 22.75/31.25 to 28.41/36.94 on DROP. This shows that explicitly modeling *how responses change* under semantic edits provides additional supervision beyond matching response endpoints alone.

4 Conclusion

We introduce S^2C, an objective-level framework for context compression that learns semantic-preserving structural invariance and structure-preserving semantic sensitivity through paired context variants. STG uses shared third-view supervision, while RSM matches edit-induced response shifts, both within the LLM's native autoregressive framework and without modifying the compression architecture. Experiments across six benchmarks show consistent gains over existing methods, especially at high compression ratios. **Limitation:** S^2C remains bounded by the capability of the underlying LLM backbone.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for [assist with translation,Generate synthetic data sets,assist in the writing of proofs,propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods]. We have not used generative AI tools for [help develop theoretical models or conceptual frameworks,formulate mathematical claims,clean and reformat dataset, support qualitative and thematic data analysis, interpret results.], and [the rest of the required disclosure tasks,provide critical ingredients for proving mathematical claims] are not applicable to this work. Additionally, we used generative AI tools for [edit a research paper to improve readability, identify relevant literature, format references]. We have reviewed all AI-assisted work. [LLM-generated code was verified and tested for correctness, all metric are checked by the authors, all words in paper are checked by the authors, all citation are checked by the authors]. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Sotiris Anagnostidis, Dario Pavllo, Luca Biggio, Lorenzo Noci, Aurelien Lucchi, and Thomas Hofmann. Dynamic context pruning for efficient and interpretable autoregressive transformers. *Advances in Neural Information Processing Systems*, 36:65202–65223, 2023.

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)*, pp. 12685–12695, 2024.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024.

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. *Advances in Neural Information Processing Systems*, 37:109487–109516, 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*, pp. 3829–3846, 2023.

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.

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. *arXiv preprint arXiv:2307.06945*, 2023.

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.

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. *arXiv preprint arXiv:2106.09685*, 2021.

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*, pp. 13358–13376, 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 (Volume 1: Long Papers)*, pp. 1658–1677, 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 (Volume 1: Long Papers)*, pp. 1601–1611, 2017.

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.

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*, pp. 6342–6353, 2023.

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)*, pp. 25081–25091, 2025.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Fanfan Liu and Haibo Qiu. Context cascade compression: Exploring the upper limits of text compression. *arXiv preprint arXiv:2511.15244*, 2025.

Xin Liu, Runsong Zhao, Pengcheng Huang, Xinyu Liu, Junyi Xiao, Tong Xiao, Shengxiang Gao, Zhengtao Yu, Jingbo Zhu, et al. Autoencoding-free context compression for llms via contextual semantic anchors. In *International Conference on Learning Representations*, volume 2026, pp. 138643–138665, 2026.

Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36:19327–19352, 2023.

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. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 963–981, 2024.

Guanghui Qin, Corby Rosset, Ethan C Chau, Nikhil Rao, and Benjamin Van Durme. Dodo: Dynamic contextual compression for decoder-only lms. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9961–9975, 2024.

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*, pp. 2383–2392, 2016.

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)*, pp. 28393–28406, 2026a.

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, pp. 46648–46666, 2026b.

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)*, pp. 28690–28704, 2026c.

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*, pp. 191–200, 2017.

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, pp. 43478–43502, 2024.

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.

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 *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

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*, volume 2025, pp. 101932–101948, 2025.

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. *arXiv preprint arXiv:2409.14364*, 2024.

A Appendix

A.1 Related Works

A.1.1 Context Compression for Large Language Models

Context compression aims to reduce the computational and memory costs of processing long inputs while retaining information required by downstream language models. Existing approaches can be broadly divided into discrete compression (Jiang et al., 2023; 2024; Pan et al., 2024) and continuous compression (Mu et al., 2023; Chevalier et al., 2023; Ge et al., 2023). Discrete methods directly reduce the number of textual tokens. LLMLingua (Jiang et al., 2023) estimates token importance and performs coarse-to-fine prompt compression, while its subsequent variants (Jiang et al., 2024; Pan et al., 2024) further improve compression for long-context and task-agnostic settings. In contrast, continuous methods encode the original context into a small number of latent representations that can be directly consumed by language models. Gist Tokens (Mu et al., 2023) modify the attention mask to encourage information to be condensed into dedicated gist tokens, enabling compressed prompts to be cached and reused. AutoCompressor (Chevalier et al., 2023) recursively summarizes long documents into compact vectors through language modeling objectives. ICAE (Ge et al., 2023) further introduces learnable memory slots and trains them with autoencoding and language modeling objectives, establishing a general paradigm for compressing arbitrary contexts into continuous memories. Beyond embedding-level memory, Activation Beacon (Zhang et al., 2025) directly compresses intermediate key-value activations at multiple transformer layers to support efficient long-context processing. Some studies (Cheng et al., 2024; Tang et al., 2026c) focus on specific application scenarios. For example, xRAG (Cheng et al., 2024) projects retrieval embeddings into the language-model representation space, representing each retrieved document using a single token for retrieval-augmented generation.

A.1.2 Learning Objectives for Context Compression

Most continuous context compression methods acquire compression capability through reconstruction or language modeling objectives, where compact representations are optimized to recover the original input or predict its continuation (Chevalier et al., 2023; Ge et al., 2023; Li et al., 2025). Such objectives provide dense supervision and do not require task-specific annotations, making them a natural choice for learning general-purpose compressed representations. However, reconstruction fidelity does not necessarily reflect the information required by downstream language understanding, especially under aggressive compression where preserving every surface-level detail becomes increasingly difficult. Recent studies have therefore begun to explore compression objectives beyond exact reconstruction. Some methods introduce downstream task supervision or relevance-aware signals to encourage compressed contexts to retain information useful for question answering and generation (Qin et al., 2024; Cheng et al., 2024). More recently, SAC (Liu et al., 2026) explicitly identifies the mismatch between autoencoding and downstream usage, and avoids reconstruction-based training by aggregating contextual information into semantic anchor tokens. Despite these advances, existing methods generally treat the original context as a unified compression target, without explicitly distinguishing semantic content from its surface realization. In contrast, our method formulates lossy compression from this distinction, encouraging invariance to semantics-preserving reformulations while retaining the response variations induced by factual changes. This enables the compressed memory to preserve information according to its functional semantic effect rather than its exact textual form.

A.2 Data Construction Prompts

A.2.1 Shared-Target Generation

For Shared-Target Generation, we use an external LLM to construct two semantically equivalent contexts with different linguistic realizations. Given an original context $X^{(1)}$, the model generates$X^{(2)}$ and $X^{(3)}$ in a single inference call using the following prompt.

> **Prompt for Shared-Target Generation**
> You are given a context. Rewrite it into two different versions that preserve exactly the same semantic content while using substantially different linguistic structures.

The two rewritten contexts must satisfy the following requirements:

1. Preserve all semantic information in the original context, including entities, attributes, facts, relations, numerical values, temporal information, conditions, and qualifications.

2. Do not add, remove, or alter any factual information.

3. Use clearly different linguistic realizations across the two versions. Vary lexical choices, sentence structures, information order, and discourse organization whenever possible.

4. Avoid superficial rewriting based only on synonym replacement. The two versions should differ meaningfully in linguistic structure while remaining semantically equivalent.

5. Preserve the level of detail of the original context. Do not summarize, shorten, elaborate, explain, or introduce information that is not present in the original.

6. Each rewritten context must be self-contained and natural.

Return only the two rewritten contexts in the following format:

[Context 1] ⟨ first rewritten context ⟩ [Context 2]

⟨ second rewritten context ⟩

Original Context: {CONTEXT}

A.2.2 RESPONSE-SHIFT MATCHING

For Response-Shift Matching, we construct a structure-preserving semantic variant conditioned on the question and its answer. Given the original context X, question Q, and answer A, the model modifies the local semantic information that determines the answer while preserving the remaining linguistic realization as much as possible.

Prompt for Response-Shift Matching

You are given a context, a question about the context, and the correct answer to that question. Modify the context so that the answer to the question changes, while keeping the linguistic structure and all unrelated information as unchanged as possible.

The edited context must satisfy the following requirements:

1. Identify the minimal piece of semantic information in the context that determines the given answer.

2. Modify only this answer-relevant information so that the correct answer to the question becomes different.

3. Keep all information unrelated to the question unchanged.

4. Preserve the original wording, sentence structure, syntax, discourse organization, and context length as much as possible. Do not paraphrase parts of the context that do not need to be changed.

5. Make the smallest possible semantic edit. Prefer changing a single entity, attribute, value, relation, date, number, or other localized fact whenever applicable.

6. The edited context must remain coherent, grammatical, and self-contained.

7. The new answer must be directly supported by the edited context and must not remain the same as the original answer.

Return the result in the following format:

[Edited Context] ⟨ edited context ⟩

[Edited Answer] ⟨ new correct answer ⟩

[Original Fact] ⟨ the minimal original fact that was changed ⟩

[Edited Fact] ⟨ the corresponding fact after editing ⟩

Context: {CONTEXT}

Question: {QUESTION}

Original Answer: {ANSWER}