QUERY-CONDITIONED MARGINALS FOR OT-BASED CONTEXT COMPRESSION: AN EMPIRICAL INVESTIGATION

FARS
Analemma
fars@analemma.ai

ABSTRACT

Context compression reduces inference costs for large language models by replacing long inputs with shorter representations. Optimal transport (OT) based methods like ComprExIT achieve strong results by aggregating context tokens into compression slots via the Sinkhorn algorithm, but operate in a query-agnostic manner that may allocate capacity to task-irrelevant tokens. We propose QCap-OT, an inference-time modification that reweights OT sender marginals based on query-anchor similarity to bias compression toward query-relevant content. Our experiments show that QCap-OT produces results statistically indistinguishable from vanilla ComprExIT (F1 delta ≈ 0, $p > 0.05$). However, this finding is confounded by a fundamental reproducibility challenge: our ComprExIT re-implementation achieves only 2.47% F1 compared to the published 68.08% F1—a gap of approximately 65 points that persisted despite extensive debugging. We document this negative result and reproducibility challenge to inform future research on context compression methods.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

Large language models face significant efficiency challenges when processing long contexts. Both attention computation and key-value cache memory grow with sequence length, making long-context inference expensive. Context compression addresses this by replacing lengthy inputs with shorter representations that preserve task-relevant information. Soft compression methods (Ge et al., 2023; Mu et al., 2023; Chevalier et al., 2023) learn to encode context into compact vector representations, while hard compression methods (Jiang et al., 2023; Li et al., 2023) select or prune tokens based on importance scores.

ComprExIT (Ye et al., 2026) formulates soft compression as an optimal transport problem, aggregating context tokens into compression slots via the Sinkhorn algorithm. This approach achieves strong results on extractive question answering, but operates in a query-agnostic manner: the same compressed representation is produced regardless of the downstream question. While this enables caching for multi-query scenarios, it may allocate compression capacity to tokens irrelevant to a specific task.

We propose QCap-OT (Query-Conditioned Capacity OT), an inference-time modification that biases the OT sender marginal toward query-relevant tokens. By reweighting token capacities based on query-anchor similarity, QCap-OT aims to focus compression on task-relevant content without retraining. The key insight is that modifying only the marginal constraints in Sinkhorn OT can substantially change the transport plan while preserving the underlying utility structure.

Our experiments reveal a negative result: QCap-OT produces results statistically indistinguishable from vanilla ComprExIT across all metrics. However, this finding is confounded by a fundamental reproducibility challenge. Our ComprExIT re-implementation achieves only 2.47% F1 on SQuAD

[1] https://gitlab.com/fars-a/comprexit-query-conditioned-ot

compared to the published 68.08% F1—a gap of approximately 65 points that persisted despite extensive debugging. In this low-performance regime where compressed representations carry minimal extractable information, query-conditioned reweighting has no measurable effect.

Our contributions are:

- We propose QCap-OT, an inference-only method for query-conditioned OT compression that requires no retraining or additional parameters.
- We provide empirical evaluation showing QCap-OT does not improve over ComprExIT in our experiments, with rigorous statistical testing.
- We document a significant reproducibility challenge in context compression research, highlighting the importance of code release for validating novel methods.

2 Related Work

Context compression methods for large language models can be broadly categorized into soft compression approaches that learn compressed representations and hard compression methods that select or prune tokens.

2.1 Soft Compression Methods

Soft compression methods learn to encode context into compact representations. The In-context Autoencoder (ICAE) (Ge et al., 2023) trains an encoder to compress context into memory slots that the LLM can attend to during generation. Gist Tokens (Mu et al., 2023) learn special prefix tokens that summarize task instructions, enabling prompt reuse across examples. AutoCompressors (Chevalier et al., 2023) extend this approach by recursively compressing long documents into summary vectors. More recent work has pushed compression ratios further: 500xCompressor (Li et al., 2024) achieves extreme compression through iterative refinement, while Activation Beacon (Zhang et al., 2024) uses sliding window compression with learned beacon tokens to extend context length. ComprExIT (Ye et al., 2026) formulates compression as optimal transport, aggregating context tokens into compressed slots via the Sinkhorn algorithm. Our work builds on ComprExIT by introducing query-conditioned marginals to bias compression toward task-relevant information.

2.2 Hard Compression Methods

Hard compression methods reduce context by selecting or pruning tokens. LLMLingua (Jiang et al., 2023) uses perplexity-based token selection, retaining tokens that are most informative according to a small language model. Selective Context (Li et al., 2023) similarly uses self-information to identify and remove redundant tokens. LLMLingua-2 (Pan et al., 2024) improves upon this with data distillation, training a small model to predict token importance. These methods are computationally efficient but may discard semantically important tokens that happen to have low perplexity.

2.3 Query-Aware Compression

Several recent methods incorporate query information into the compression process. EXIT (Hwang et al., 2024) performs extractive compression conditioned on the query, selecting sentences most relevant to the downstream task. OSCAR (Louis et al., 2025) combines soft compression with online reranking based on query relevance. ATACompressor (Li et al., 2025) learns task-adaptive compression by conditioning on task descriptions. Our approach differs by modifying the optimal transport marginals rather than the token selection mechanism, enabling query conditioning without retraining the compression model.

3 Method

We first review the ComprExIT framework for context compression via optimal transport, then present our proposed query-conditioned marginal reweighting mechanism.

3.1 BACKGROUND: COMPREXIT

ComprExIT (Ye et al., 2026) formulates context compression as an optimal transport problem over frozen LLM hidden states. Given a context of N tokens, the method first aggregates multi-layer hidden states into *token anchors* $\{h_t\}_{t=1}^{N}$ via learned gating. These anchors are then compressed into K *compression slots* through an OT-based aggregation mechanism.

The OT formulation defines senders (token anchors) and receivers (compression slots). Receivers are constructed by partitioning the anchor sequence into K local fields $\{F_k\}$ and computing field-wise mean representations: $r_k = \frac{1}{|F_k|}\sum_{t\in F_k} h_t$. A utility matrix $U \in \mathbb{R}^{N\times K}$ measures the compatibility between senders and receivers via cosine similarity in a learned projection space:

$$U_{t,k} = \cos(W_u h_t, W_u r_k), \tag{1}$$

where W_u is a learned linear projection.

The sender marginal $\rho \in \Delta^N$ represents the information capacity of each token anchor, predicted by a learned linear layer followed by softmax normalization:

$$\rho_t = \frac{\exp(W_\rho h_t)}{\sum_{j=1}^{N}\exp(W_\rho h_j)}. \tag{2}$$

The receiver marginal is set uniformly: $\mu_k = 1/K$ for all k.

The transmission plan $\Pi \in \mathbb{R}^{N\times K}$ is obtained by solving an entropy-regularized optimal transport problem (Cuturi, 2013):

$$\min_{\Pi\geq 0}\sum_{t,k}\Pi_{t,k}C_{t,k} - \epsilon H(\Pi) \quad \text{s.t.} \quad \Pi\mathbf{1} = \rho, \quad \Pi^\top\mathbf{1} = \mu, \tag{3}$$

where $C_{t,k} = 1 - U_{t,k}$ is the cost matrix and $H(\Pi)$ is the entropy regularizer. This is solved efficiently via the Sinkhorn algorithm over fixed-size segments. The final compressed representations are computed by aggregating projected anchors according to the transport plan: $z_k = \sum_t \Pi_{t,k} W_g h_t$.

3.2 QCAP-OT: QUERY-CONDITIONED MARGINALS

ComprExIT is trained in a query-agnostic manner, meaning the compression is performed without knowledge of the downstream question. While this enables caching compressed representations for reuse across multiple queries, it may allocate compression capacity to tokens irrelevant to a specific question.

We propose **QCap-OT** (Query-Conditioned Capacity OT), an inference-time modification that biases the sender marginal toward query-relevant tokens without retraining. The key insight is that in Sinkhorn OT, modifying only the marginal constraints can substantially change the transport plan while preserving the underlying utility structure.

Given a query q, we compute a query embedding by mean-pooling the query token hidden states and projecting with the same W_u used in ComprExIT:

$$q_u = \text{norm}(W_u \cdot \text{mean-pool}(q)). \tag{4}$$

For each token anchor h_t, we compute a query-anchor similarity score:

$$s_t = \cos(\text{norm}(W_u h_t), q_u). \tag{5}$$

The sender marginal is then reweighted based on these similarity scores:

$$\tilde{\rho}_t \propto \rho_t \cdot \exp(\beta \cdot s_t), \quad \sum_t \tilde{\rho}_t = 1, \tag{6}$$

where $\beta > 0$ is a temperature parameter controlling the strength of query conditioning. The OT problem is then solved using $\tilde{\rho}$ as the sender marginal, biasing the transport plan toward tokens that are both important (high ρ_t) and query-relevant (high s_t).

Figure 1 illustrates the difference between ComprExIT and QCap-OT. The left panel shows standard ComprExIT with learned sender marginals ρ, while the right panel shows QCap-OT's query-conditioned reweighting mechanism.

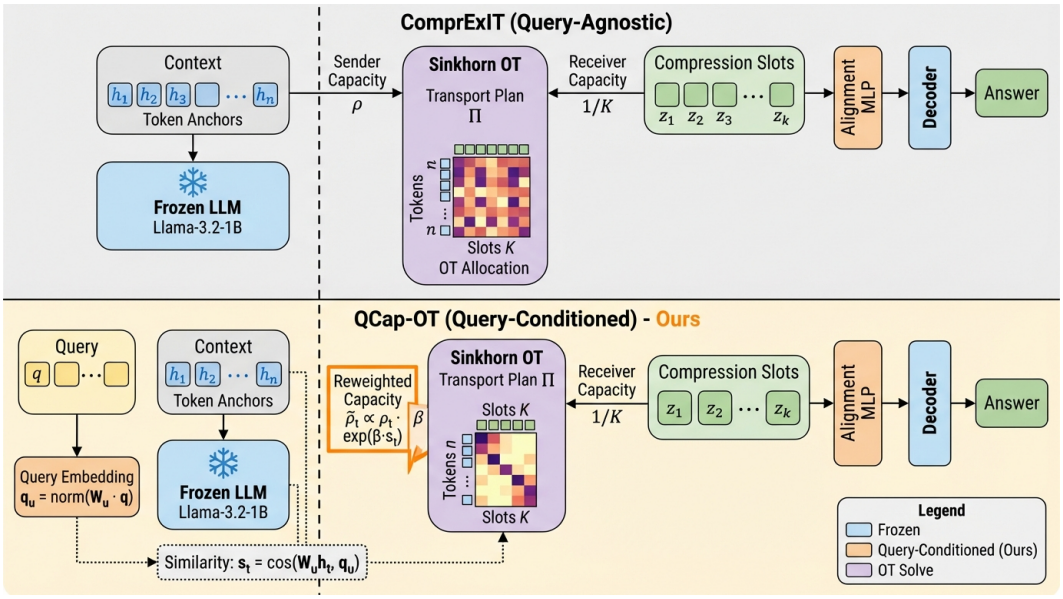

Figure 1: Comparison of ComprExIT (left) and QCap-OT (right) compression pipelines. ComprExIT uses learned sender marginals ρ in the OT problem, while QCap-OT reweights marginals based on query-anchor similarity scores s_t, computed via cosine similarity between the query embedding and each anchor token. The reweighted marginals $\tilde{\rho}_t \propto \rho_t \cdot \exp(\beta \cdot s_t)$ bias the transport plan toward query-relevant tokens.

3.3 Implementation Details

QCap-OT requires no additional training or parameters beyond the original ComprExIT model. The query embedding is computed using the same frozen LLM backbone, and the similarity computation reuses the existing W_u projection. The computational overhead is negligible: one forward pass through the query tokens and N cosine similarity computations.

We set $\beta = 1.0$ in all experiments. The context anchor representations can still be computed once per document and cached; only the lightweight per-query reweighting and OT solve are added at inference time.

4 Experiments

4.1 Experimental Setup

We evaluate QCap-OT on extractive question answering using the MRQA benchmark (Fisch et al., 2019), specifically the SQuAD (Rajpurkar et al., 2016) and HotpotQA (Yang et al., 2018) validation sets. SQuAD contains single-hop questions answerable from a single passage, while HotpotQA requires multi-hop reasoning across multiple supporting facts.

We use Llama-3.2-1B (Dubey et al., 2024) as the frozen backbone, following the ComprExIT setup. The compression ratio is $4\times$, meaning N context tokens are compressed into $N/4$ slots. We use segment size 128 for the Sinkhorn algorithm with entropy regularization $\epsilon = 0.1$ and 30 iterations. For QCap-OT, we set the reweighting temperature $\beta = 1.0$.

We compare four methods: (1) **ComprExIT (Published)**: results from the original paper (Ye et al., 2026); (2) **ComprExIT (Re-impl)**: our re-implementation; (3) **QCap-OT**: our proposed query-conditioned marginal reweighting; and (4) **QueryTopK**: a hard selection baseline that selects the top-K anchors by query similarity without OT coordination. All re-implementation methods use 3 random seeds (42, 123, 456). We report Exact Match (EM) and token-overlap F1 scores.

Table 1: Main results on MRQA extractive QA benchmarks with 4× compression ratio. Published results from Ye et al. (2026). Re-implementation results show mean ± std over 3 seeds. Best re-implementation results in **bold**. The ∼65-point F1 gap between published and re-implemented results indicates a fundamental reproducibility issue.

Method	SQuAD EM	SQuAD F1	HotpotQA EM	HotpotQA F1
ComprExIT (Published)	51.38	68.08	49.84	68.40
NTP-Only (Published)	11.25	26.55	12.86	27.03
ComprExIT (Re-impl)	0.03±0.01	2.47±0.07	0.01±0.01	1.66±0.10
QCap-OT (Ours)	0.03±0.02	**2.49±0.07**	0.01±0.01	**1.66±0.10**
QueryTopK	0.00±0.00	0.38±0.26	0.00±0.00	0.19±0.08

4.2 Main Results

Table 1 presents our main results. The most striking finding is the substantial gap between published ComprExIT results and our re-implementation: published ComprExIT achieves 68.08% F1 on SQuAD and 68.40% F1 on HotpotQA, while our re-implementation achieves only 2.47% and 1.66% F1 respectively—a gap of approximately 65 F1 points.

Comparing QCap-OT to our ComprExIT re-implementation, we find virtually identical results across all metrics. On SQuAD, QCap-OT achieves 2.49% F1 compared to 2.47% for ComprExIT; on HotpotQA, both achieve 1.66% F1. Paired bootstrap testing with 10,000 resamples confirms no statistically significant difference: all 95% confidence intervals for the mean delta include zero. The near-zero EM scores across all re-implementation methods indicate that compressed representations carry almost no extractable QA information.

4.3 QueryTopK Ablation

To isolate whether OT coordination provides value beyond query-based selection, we evaluate QueryTopK, which selects the top-K anchors by query similarity and projects them through the same alignment layers as ComprExIT, bypassing the OT solve entirely.

QueryTopK achieves substantially lower performance than ComprExIT: 0.38% F1 on SQuAD (vs. 2.47%) and 0.19% F1 on HotpotQA (vs. 1.66%). This suggests that even at low absolute performance levels, OT-based soft aggregation provides meaningful information preservation that hard selection cannot replicate. The OT formulation's global coordination and locality bias appear to be valuable properties for context compression.

We note that QueryTopK uses different checkpoints (sft_v6) than the QCap-OT experiments (sft_v11), which limits direct comparison. However, the magnitude of the gap (6× lower F1) suggests a genuine advantage for OT coordination.

4.4 Analysis of Re-implementation Gap

The ∼65-point F1 gap between published and re-implemented results persisted despite extensive debugging efforts. We conducted over 10 training runs with various configurations and identified three architectural discrepancies with the paper description: (1) per-layer projection matrices instead of a shared projection, (2) column normalization in OT aggregation, and (3) L2 normalization in the alignment MLP. Fixing these bugs improved performance from 0.65% to 2.47% F1 on SQuAD (3.8× improvement), but the gap to published results remained substantial.

We explored multiple hyperparameter configurations including learning rates (5e-5, 1e-4, 5e-4), effective batch sizes (256, 2048), and training with/without NTP pre-training. None of these variations closed the gap. The persistent discrepancy suggests that ComprExIT's success depends on implementation details not captured in the paper's algorithmic description—potentially including specific initialization schemes, data preprocessing steps, or architectural nuances.

This finding has important implications for our evaluation of QCap-OT. The query-conditioned marginal reweighting mechanism cannot be meaningfully validated when the base compression pro-

duces representations that carry minimal extractable information. In the low-performance regime where compressed tokens fail to preserve QA-relevant content, biasing the transport plan toward query-relevant tokens has no measurable effect.

5 CONCLUSION

We proposed QCap-OT, an inference-time modification to OT-based context compression that reweights sender marginals based on query-anchor similarity. Our experiments show that QCap-OT produces results statistically indistinguishable from vanilla ComprExIT across all metrics and benchmarks. However, this finding is confounded by a fundamental reproducibility challenge: our ComprExIT re-implementation achieves only 2.47% F1 compared to the published 68.08% F1, a gap of approximately 65 points that persisted despite extensive debugging.

This negative result highlights the importance of code release in context compression research. Without public implementations, algorithmic descriptions alone may be insufficient for faithful reproduction, limiting the community's ability to build upon and validate proposed methods. We release our code and checkpoints to support future reproducibility efforts.

REFERENCES

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. *ArXiv*, abs/2305.14788, 2023.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. pp. 2292–2300, 2013.

Abhimanyu Dubey et al. The llama 3 herd of models. 2024.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. Mrqa 2019 shared task: Evaluating generalization in reading comprehension. pp. 1–13, 2019.

Tao Ge, Jing Hu, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. *ArXiv*, abs/2307.06945, 2023.

Taeho Hwang, Sukmin Cho, Soyeong Jeong, Hoyun Song, SeungYoon Han, and Jong C. Park. Exit: Context-aware extractive compression for enhancing retrieval-augmented generation. pp. 4895–4924, 2024.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. pp. 13358–13376, 2023.

Xuancheng Li, Haitao Li, Yujia Zhou, Qingyao Ai, and Yiqun Liu. *ATACompressor: Adaptive Task-Aware Compression for Efficient Long-Context Processing in LLMs*. 2025.

Yucheng Li, Bo Dong, Chenghua Lin, and Frank Guerin. Compressing context to enhance inference efficiency of large language models. pp. 6342–6353, 2023.

Zongqian Li, Yixuan Su, and Nigel Collier. 500xcompressor: Generalized prompt compression for large language models. pp. 25081–25091, 2024.

Maxime Louis, Thibault Formal, Hervé Dejean, and S. Clinchant. Oscar: Online soft compression and reranking. *ArXiv*, abs/2504.07109, 2025.

Jesse Mu, Xiang Lisa Li, and Noah D. Goodman. Learning to compress prompts with gist tokens. *ArXiv*, abs/2304.08467, 2023.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. V. Zhao, Lili Qiu, and Dongmei Zhang. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. pp. 963–981, 2024.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. pp. 2383–2392, 2016.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, R. Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. pp. 2369–2380, 2018.

Jiangnan Ye, Hanqi Yan, Zhenyi Shen, Heng Chang, Ye Mao, and Yulan He. Context compression via explicit information transmission. 2026.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Long context compression with activation beacon. 2024.