Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Cross-Modality Compression Efficiency and Reconstruction Fidelity in Mobile Cloud Computing: A Unified Runtime Analysis of the EADC Framework

Victoria Boafo

victoria.boafo@mtce.edu.gh

Mampong Technical College of Education https://orcid.org/0000-0002-7783-1086

Research Article

Keywords: Mobile cloud computing, Cross-modality compression, Near-lossless compression, Runtime decomposition, Reconstruction accuracy, Throughput analysis

Posted Date: November 11th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11683539/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Abstract

Mobile cloud computing backends must compress heterogeneous data under strict latency and bandwidth constraints, yet compression studies usually evaluate one data type and report runtime only as an aggregate. This paper presents a unified runtime and reconstruction-fidelity evaluation of the Enhanced Adaptive Discriminant Compression framework across its image and text pathways. An additive runtime model attributes processing time to kernel and discriminant computation, dictionary-based redundancy elimination, and adaptive parameter updates. Experiments on eight image categories from the Stanford Light Field Raw Image Dataset and eight text collections from the Pizza & Chili Repetitive Corpus show reconstruction accuracy between 99.90% and 99.98%. Image-pathway runtime differences against dictionary and context-model baselines are statistically significant ($p < .001$). Mean total runtime is 3.71 s per image and 8.01 s per text collection, corresponding to throughputs of 11.2 MB/s and 58.4 MB/s, which supports near-real-time deployment on cloud backends.

1 Introduction

Mobile cloud computing (MCC) offloads storage- and compute-intensive tasks from resource-constrained mobile devices to cloud backends [1, 2], and the volume and heterogeneity of the data crossing that link (images, sensor logs and free text) continue to grow, placing increasing storage and energy pressure on cloud infrastructure [3]. Because compression sits directly on the critical path between device and backend, it mediates both the transmission bandwidth consumed and the backend latency incurred before downstream processing can begin [4–7]. Frameworks that report only a compression ratio, without a runtime model that a backend engineer can use for capacity planning, leave an important deployment question open: how much wall-clock cost does a unified compression scheme add per data type, and where does that cost originate?

Most compression evaluations validate a single modality in isolation: dictionary-based methods such as LZ77 [4] and its learned or grammar-based derivatives [8, 9] for byte streams; probabilistic context models such as prediction by partial matching (PPM) and back-off estimators [5, 6] for text; and transform-, kernel- or network-based codecs [10–14] for images. Hybrid and cloud-oriented compression studies [7, 15–17] typically report one aggregate runtime figure without decomposing it into the algorithmic stages that produced it. This makes it difficult to reason about where a unified, multi-component framework spends its computational budget when it serves mixed image and text workloads on the same cloud cluster.

This paper develops a runtime-focused evaluation of the Enhanced Adaptive Discriminant Compression (EADC) framework developed in the authors' doctoral research [18]. EADC is a unified maximum a posteriori (MAP) objective that integrates Gaussian kernel-based nonlinear reconstruction, discriminant regularization, LZ77 dictionary coding and PPM/N-gram contextual modelling within a single dual-pathway architecture. The doctoral work established the framework and its compression performance;

here we focus on its systems-level behaviour, namely runtime efficiency and reconstruction fidelity evaluated jointly across the image and text pathways.

The contributions of this paper are as follows:

(1) An additive runtime decomposition model, $T_{EADC} = T_{kernel} + T_{LZ77} + T_{Adaptive}$, that attributes total EADC runtime to kernel and discriminant computation, dictionary search and adaptive regularization-parameter updates.

(2) A cross-modality reconstruction-accuracy evaluation spanning image and text datasets, showing a band of only 0.08% (99.90–99.98%) across all evaluated files.

(3) One-way analysis of variance (ANOVA) of runtime differences among LZ77, PPM and EADC on the image pathway ($F(2, 21) = 14.62$, $p < .001$, $\eta^2 = 0.58$), together with descriptive runtime analysis across eight text collections.

(4) A representative text-runtime comparison showing 24.8% and 11.8% total-runtime reductions for EADC relative to LZ77 and PPM, respectively.

(5) A deployment-suitability assessment showing throughputs of 11.2 MB/s (image) and 58.4 MB/s (text), and a discussion of how the decomposition supports asymmetric provisioning of cloud cluster resources.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 presents the system model, runtime decomposition, datasets and statistical methodology. Section 4 reports the reconstruction-accuracy, runtime and statistical results. Section 5 discusses the implications for cloud backend deployment and the limitations of the study, and Section 6 concludes.

2 Related work

Redundancy-elimination methods rooted in LZ77 [4] and its space-efficient grammar-based derivatives [9] remain the default building block for general-purpose byte-stream compression, and machine-learning-assisted variants have been proposed to identify and restore LZ77-compressed segments [8]. On the probabilistic side, PPM [5] and back-off frequency estimators [6] exploit sequential context to compress natural-language and symbolic text. Dictionary-learning formulations [19] generalize these ideas to structured, supervised representations.

For image data, transform-based standards such as JPEG-LS [10] and JPEG 2000 [11] remain widely deployed, while learned codecs [12–14] and near-lossless remote-sensing schemes [17] have pushed reconstruction fidelity closer to the original signal at higher computational cost. Error-bounded modular frameworks such as SZ3 [20] similarly trade a configurable fidelity budget for compression gain in scientific-data settings. Across both modalities, however, published runtime figures are almost always

reported as a single aggregate wall-clock number, without a decomposition that separates algorithmic stages.

Closer to the deployment question addressed here, cloud- and edge-oriented studies have examined adaptive compression for cloud-hosted 3D healthcare data [15], reconfigurable-intelligent-surface-aided mobile edge computing [16] and surveys of data compression techniques and their applications [7]; a recent review surveys distributed, learning-based compression relevant to MCC-style pipelines [21], and scalable medical-image compression [22] targets bandwidth-constrained transmission. None of these studies jointly (i) decomposes a unified compression framework's runtime into its constituent computational stages and (ii) statistically examines the throughput implications of that decomposition across two structurally different data modalities under identical hardware and cross-validation conditions. This is the gap this paper addresses.

Recent work in mobile cloud and edge computing has concentrated mainly on where computation is placed. Sanaei et al. [1] characterized heterogeneity as a central challenge of MCC, and Mora et al. [2] generalized MCC to offload advanced Internet of Things applications across several network computing layers. Moradi and Rezaei [23] proposed task caching for mobile edge computing, and Sahkhar et al. [24] optimized cloudlet placement to increase device coverage and reduce network access delay. These approaches decide where and when tasks run, but they treat the volume of data exchanged between device and backend as given. Compression reduces that volume directly, so its runtime cost should be considered in the same offloading and provisioning decisions.

3 System model and methodology

3.1 EADC dual-pathway architecture

EADC solves a single MAP optimization problem that integrates Gaussian kernel-based nonlinear reconstruction, discriminant regularization, LZ77-based redundancy elimination and PPM/N-gram contextual modelling (Fig. 1). Input is routed through an image pathway or a text pathway according to its format, while both pathways share the same discriminant-regularized objective and block-coordinate optimization procedure. Because the image pathway operates on high-dimensional pixel matrices and the text pathway on one-dimensional symbol streams, the two pathways activate the shared regularization terms λ_1–λ_4 to different degrees. As Section 4 shows, this produces different runtime profiles even though reconstruction fidelity remains stable across both.

3.2 Additive runtime decomposition model

To make the computational cost of EADC analysable at the systems level, total processing time is defined as the sum of the compression and decompression stages,

$$T_{\text{total}} = T_{\text{compress}} + T_{\text{decompress}} \quad (1)$$

and the EADC-specific runtime is further decomposed additively across its three computational stages,

$$T_{\text{EADC}} = T_{\text{kernel}} + T_{\text{LZ77}} + T_{\text{Adaptive}} \quad (2)$$

where T_{kernel} is the time spent on kernel mapping and discriminant optimization, T_{LZ77} is the time spent on dictionary search and tuple generation, and T_{Adaptive} is the time spent on λ_1–λ_4 parameter updates. This additive structure reflects separable computational costs and lets a systems engineer attribute observed throughput to a specific algorithmic stage rather than treating runtime as a single aggregate value. Sections 4 and 5 interpret the results using this model.

3.3 Datasets

The image pathway was evaluated on the Stanford Light Field Raw Image Dataset using eight semantic categories (People, Buildings, Cars, Bikes, Flowers, Fruits, General and Occlusion). Each category is represented by one class-representative instance standardized to 5368 × 7728 pixels (41,483,904 pixels; approximately 41.48 MB per single-channel raw image), giving a total experimental image volume of approximately 331.9 MB.

The text pathway was evaluated on the Pizza & Chili Repetitive Corpus using eight collections covering natural-language documents, genomic sequences and highly repetitive symbolic data. Their original sizes are: Einstein.en.txt, 467,626,544 bytes; Cere, 461,373,440 bytes; Para, 429,916,160 bytes; Influenza, 155,189,248 bytes; Escherichia coli, 113,246,208 bytes; Protein, 104,857,600 bytes; Einstein.de.txt, 93,323,264 bytes; and World Leaders, 47,185,920 bytes. Einstein.en.txt is retained as the representative case for the LZ77/PPM/EADC runtime comparison in Section 4.3.

3.4 Experimental protocol and computing environment

Each dataset was divided into exclusive training (60%), validation (20%) and testing (20%) subsets. The training subset was used to learn dictionary representations, kernel mappings and discriminant coefficients; the validation subset was used for hyperparameter selection; and the held-out test subset was reserved for final compression-ratio, reconstruction-accuracy, runtime and statistical evaluation. Ten-fold cross-validation was applied at the segment level: high-resolution images were partitioned into spatial patches, and text files were divided into sequential symbol blocks. This procedure reduces information leakage during parameter selection and evaluation.

All experiments were executed on a workstation with an Intel Core i7 CPU, 16 GB RAM and an NVIDIA RTX GPU, which was used for MATLAB-supported parallel acceleration of the kernel-mapping and discriminant-learning stages. EADC and all baseline methods (Portable Network Graphics (PNG), the JBIG2 bi-level image coder, LZ77 and PPM) were implemented in MATLAB R2024b using the Image Processing Toolbox and the Statistics and Machine Learning Toolbox for kernel computation, optimization and inferential statistics.

3.5 Evaluation metrics

Because the EADC image pathway performs kernel-based numerical reconstruction rather than strict symbolic bitwise reconstruction, the framework is evaluated in a near-lossless regime. A single normalized fidelity measure is therefore applied to both image matrices and text representations, rather than modality-specific image measures such as peak signal-to-noise ratio or structural similarity. Reconstruction accuracy (RA) is the percentage of matching bytes retained after decompression relative to the original file size,

RA (%) = (matching bytes after decompression / original file size) × 100 (3)

and runtime efficiency is the total runtime defined in Eq. (1). Both metrics are computed identically for the image and text pathways, so the two pathways can be compared directly in Section 4.

3.6 Hyperparameter selection and optimization

The regularization parameters λ_1–λ_4 were selected by cross-validation over the range 0.01–1.00 with a step size of 0.01. The resulting pathway-specific settings are listed in Table 1. For the representative Einstein.en.txt experiment, the Gaussian-kernel bandwidth was σ = 0.12. Optimization uses alternating block-coordinate updates consistent with the proximal alternating linearized minimization (PALM) framework [25]: the coefficient matrix is updated by regularized gradient steps with proximal stabilization, and the dictionary matrix is then updated while preserving its column-normalization constraints. The objective is evaluated after each iteration, and optimization terminates when the prescribed objective-decrease condition is satisfied.

Table 1
Regularization hyperparameters by pathway

Pathway	λ_1	λ_2	λ_3	λ_4
Image	0.01	0.02	0.00	0.00
Text	0.05	0.03	0.04	0.02

3.7 Statistical procedures

Runtime differences among LZ77, PPM and EADC on the image pathway were evaluated with one-way ANOVA at the 95% confidence level, with three algorithms evaluated across eight image categories ($df_{between} = 2$, $df_{within} = 21$). Effect size is reported as $\eta^2 = (F \cdot df_{between}) / (F \cdot df_{between} + df_{within})$. Text-pathway runtime is reported descriptively across the eight collections, because an inferential text-runtime result consistent with the eight-collection design was not available (Online Resource 1, Sect. S7).

4 Results

4.1 Cross-modality reconstruction accuracy

Table 2 reports post-decompression reconstruction accuracy for representative image and text files, computed on the evaluated byte count of each file. All evaluated files, regardless of modality, achieve RA between 99.90% and 99.98%, a band of only 0.08%. The lowest value (99.90%, Escherichia coli) is attributable to the higher intrinsic entropy and lower redundancy of genomic sequence data, yet it remains within near-lossless bounds. This narrow band indicates that the unified EADC objective preserves local and global data structure whether the input is a two-dimensional pixel matrix or a one-dimensional symbol stream.

Table 2
Cross-modality reconstruction accuracy after decompression

Dataset / file	Modality	Evaluated size (bytes)	Matching bytes after decompression	RA (%)
People	Image	41,483,904	41,473,000	99.97
Buildings	Image	41,483,904	41,475,500	99.98
Cars	Image	41,483,904	41,471,000	99.97
Cere	Text	461,286,644	461,180,000	99.98
Para	Text	429,270,000	429,155,000	99.97
Escherichia coli	Text	112,690,000	112,579,000	99.90

RA reconstruction accuracy. For the text collections, RA is computed on the byte stream evaluated by the text pathway; these evaluated sizes are slightly smaller than the original file sizes listed in Sect. 3.3

4.2 Runtime: image pathway

Table 3 reports per-category compression time, decompression time and total runtime for the image pathway. Total runtime ranges from 3.4 s (Flowers) to 4.0 s (General), with a mean of 3.71 s across the eight categories, corresponding to an average throughput of approximately 11.2 MB/s. Because the image pathway primarily activates the kernel-reconstruction (λ_1) and LZ77-redundancy (λ_2) terms, its runtime remains low relative to the size of the images processed, which supports near-real-time use in an MCC backend.

Table 3
EADC runtime on the image pathway (Stanford Light Field Raw Image Dataset)

Category	Compression ratio	Compression time (s)	Decompression time (s)	Total runtime (s)
People	11.35	2.5	1.2	3.7
Buildings	14.86	2.6	1.3	3.9
Cars	10.45	2.4	1.2	3.6
Bikes	10.89	2.5	1.2	3.7
Flowers	9.49	2.3	1.1	3.4
General	10.85	2.7	1.3	4.0
Fruits	14.75	2.4	1.1	3.5
Occlusion	9.79	2.6	1.3	3.9
Mean	**11.55**	**2.50**	**1.21**	**3.71**

Mean throughput ≈ 11.2 MB/s (41.48 MB per image / 3.71 s)

4.3 Runtime: text pathway

Table 4 compares LZ77, PPM and EADC on the representative Einstein.en.txt collection. EADC completes compression and decompression in 8.2 s in total, against 10.9 s for LZ77 and 9.3 s for PPM: reductions of 24.8% and 11.8%, respectively. Table 5 extends the evaluation to all eight collections, giving a mean total runtime of 8.01 s. Taking the Einstein.en.txt file size as the representative workload, this mean corresponds to an indicative throughput of approximately 58.4 MB/s. Although all four regularization terms are active on the text pathway, EADC remains efficient because its PPM and N-gram components jointly reduce unnecessary context evaluations, which keeps the T_{Adaptive} term of Eq. (2) small.

Table 4
Runtime comparison for Einstein.en.txt (representative text case)

Method	Compression time (s)	Decompression time (s)	Total runtime (s)
LZ77	7.4	3.5	10.9
PPM	6.2	3.1	9.3
EADC	5.4	2.8	8.2
EADC vs. LZ77: 24.8% reduction; EADC vs. PPM: 11.8% reduction			

Table 5
EADC runtime on the Pizza & Chili Repetitive Corpus text collections

Collection	Compression ratio	Compression time (s)	Decompression time (s)	Total runtime (s)
Cere	16.37	5.2	2.7	7.9
Para	19.84	4.8	2.6	7.4
Escherichia coli	14.43	5.5	2.8	8.3
Einstein.de.txt	13.91	5.3	2.9	8.2
Einstein.en.txt	12.51	5.4	2.8	8.2
Influenza	13.87	5.6	2.9	8.5
World Leaders	11.81	5.0	2.7	7.7
Protein	14.85	5.1	2.8	7.9
Mean	**14.70**	**5.24**	**2.78**	**8.01**

Indicative throughput ≈ 58.4 MB/s (467,626,544 bytes / 8.01 s)

4.4 Statistical validation

Table 6 summarizes the inferential result. The image-pathway one-way ANOVA gives $F(2, 21) = 14.62$, $p < .001$, with a large effect size ($\eta^2 = 0.58$), indicating that the choice among LZ77, PPM and EADC explains more than half of the observed variance in image runtime. Text-pathway runtime is summarized descriptively in Table 5.

Table 6
Statistical validation summary

Test	Comparison	Statistic	*df*	*p*-value	Effect size
One-way ANOVA	Image pathway: LZ77 vs. PPM vs. EADC	$F = 14.62$	2, 21	< .001	$\eta^2 = 0.58$

η^2 computed from the reported *F* statistic and degrees of freedom

4.5 Compression ratio and component ablation

The mean EADC compression ratio is 11.55 across the eight image categories (Table 3) and 14.70 across the eight text collections (Table 5). For the image pathway, the doctoral research [18] reports baseline ratios of 10.96 for PNG, 9.54 for LZ77 and 9.29 for JBIG2. For Einstein.en.txt (467,626,544 bytes), the reported compression ratios are 7.59 for LZ77, 11.57 for PPM and 12.51 for EADC, giving an

EADC compressed size of 37,380,219 bytes. These results show that the runtime advantage is not obtained at the expense of compression efficiency.

An image-pathway ablation study removed individual components while keeping all other experimental conditions and hyperparameters fixed. The complete EADC pathway achieved an average compression ratio of 11.55. Removing the kernel reconstruction term (EADC–K) reduced it to 9.84 (14.8% reduction), removing discriminant regularization (EADC–D) reduced it to 10.27 (11.1% reduction), and removing the LZ77 redundancy term (EADC–LZ) reduced it to 8.91 (22.9% reduction). Each component therefore contributes to the overall compression ratio, and no single module accounts for the performance.

4.6 Deployment suitability

Table 7 consolidates the deployment-relevant figures. EADC sustains approximately 11.2 MB/s for images and an indicative 58.4 MB/s for text. The image runtime difference is statistically significant, while text runtime is reported descriptively. These figures indicate that EADC can serve as a near-real-time compression stage for mixed image and text MCC workloads without a modality-specific fallback.

Table 7
Deployment suitability summary

Metric	Image pathway	Text pathway
Mean total runtime	3.71 s	8.01 s
Mean throughput	11.2 MB/s	≈ 58.4 MB/s
Runtime reduction vs. LZ77	n/a	24.8% (Einstein.en.txt)
Runtime reduction vs. PPM	n/a	11.8% (Einstein.en.txt)
Reconstruction accuracy range	99.97–99.98%	99.90–99.98%
Cross-modality RA band	0.08%	0.08%

n/a not applicable; the LZ77 and PPM runtime comparison was performed on the text pathway

5 Discussion

5.1 Implications for cloud backend provisioning

The additive decomposition in Eq. (2) offers a systems-level explanation for the throughput gap between pathways. On the image pathway, T_{kernel} dominates because Gaussian kernel mapping and discriminant optimization operate over large, dense pixel matrices (≈ 41.48 MB per image), while only the λ_1 and λ_2 terms are strongly activated; this keeps runtime low relative to file size but limits throughput to about 11.2 MB/s. On the text pathway, all four terms are active, yet PPM and N-gram context modelling operate over a compact symbol alphabet with joint context reduction, so T_{Adaptive} contributes proportionally less

overhead per byte than T_{kernel} does on the image side. This is consistent with the higher text throughput of approximately 58.4 MB/s.

For MCC backend provisioning, the decomposition suggests that EADC should not be treated as a single, uniform cost. Image-heavy workloads are bounded mainly by kernel and discriminant computation and would benefit most from GPU-accelerated kernel evaluation, whereas text-heavy workloads are bounded mainly by adaptive context-model updates and would benefit most from optimized context-table lookups. A cluster that serves a mixed workload can therefore provision compute asymmetrically, for example by placing image-pathway jobs on GPU-equipped nodes and text-pathway jobs on CPU-dense nodes. Because individual files are compressed independently, the pathways can also be distributed across cluster nodes in parallel; measuring the resulting scaling, including input/output and network contention, remains future work.

The 0.08% cross-modality reconstruction-accuracy band is also a key systems result. It shows that a single unified objective keeps near-lossless fidelity constant while runtime varies with modality. As a result, the accuracy of EADC does not need to be re-evaluated for each data type; only its runtime depends on the pathway used. The ablation results reinforce this interpretation, since the components that contribute to compression efficiency also correspond to identifiable computational stages in the runtime model.

5.2 Limitations and reproducibility

These results should be read with three limitations in mind. First, all measurements were obtained on a single workstation-class configuration (Section 3.4); absolute throughput will differ on production cloud instances and multi-node clusters, although the relative decomposition and statistical comparisons are expected to hold. Second, the MATLAB implementation is a research prototype rather than a production runtime, so the reported throughput figures should be treated as a lower bound on what an optimized, compiled deployment could achieve; moreover, performance and cost differ across cloud instance types [26], so deployment results will depend on instance selection. Third, the evaluation covers eight image categories and eight text collections; other data types, such as audio and sensor time series, are left to future work. Consequently, absolute runtime values should be interpreted as measurements under the stated set-up, whereas the component-wise interpretation and relative comparisons are expected to transfer to other platforms.

6 Conclusion

This paper presented a unified, cross-modality runtime and reconstruction-fidelity evaluation of the EADC compression framework for mobile cloud computing. The additive runtime model, $T_{\text{EADC}} = T_{\text{kernel}} + T_{\text{LZ77}} + T_{\text{Adaptive}}$, attributes total processing time to interpretable computational stages and explains the throughput difference between the image pathway (11.2 MB/s) and the text pathway (approximately 58.4 MB/s). Reconstruction accuracy remained within a 0.08% band (99.90–99.98%) across both modalities, and one-way ANOVA on the image pathway ($F(2, 21) = 14.62$, $p < .001$, $\eta^2 = 0.58$) confirmed a

significant runtime difference among LZ77, PPM and EADC. The results indicate that EADC can operate as a near-real-time backend compression service for mixed image and text workloads, and the runtime model provides a basis for asymmetric capacity planning across data modalities. Future work will validate throughput scaling on multi-node cloud clusters and extend the evaluation to further data types.

Declarations

Acknowledgements

The authors thank the Department of Computer Science, School of Computing and Information Sciences, University of Technology and Applied Sciences, Ghana, for providing the computational resources used in this study.

Funding The authors declare that no funds, grants or other support were received during the preparation of this manuscript.

Competing interests The authors have no relevant financial or non-financial interests to disclose.

Author contributions Victoria Boafo: conceptualization, methodology, software, formal analysis, investigation, data curation, writing – original draft. Edward Y. Baagyere: supervision, validation, writing – review and editing. Paula Aninyie Wumnaya: supervision, validation, writing – review and editing. All authors read and approved the final manuscript.

Data availability The datasets analysed in this study are publicly available. The Stanford Light Field Raw Image Dataset is available at https://lightfields.stanford.edu/LF2016.html, and the Pizza & Chili Repetitive Corpus is available at http://pizzachili.dcc.uchile.cl/repcorpus.html. The experimental protocol, dataset scope and parameter settings are provided in Online Resource 1.

Code availability The MATLAB implementation of EADC is available from the corresponding author on reasonable request.

Ethics approval Not applicable. This study uses only publicly available benchmark datasets and involves no human participants or animals.

Consent to participate and consent for publication Not applicable.

Use of generative AI Generative AI tools were used to assist with language editing and restructuring of the authors' own text. The authors reviewed and verified all content and take full responsibility for it.

References

1. Sanaei Z, Abolfazli S, Gani A, Buyya R (2014) Heterogeneity in mobile cloud computing: taxonomy and open challenges. IEEE Commun Surv Tutor 16(1):369–392

2. Mora H, Pujol FA, Ramírez T, Jimeno-Morenilla A, Szymanski J (2024) Network-assisted processing of advanced IoT applications: challenges and proof-of-concept application. Cluster Comput 27:1849–1865. https://doi.org/10.1007/s10586-023-04050-6
3. Xu X, Wang R, Zhang Y (2025) Optimization analysis of scheduling strategy for cloud storage architecture based on hierarchical service mechanism. Cluster Comput 28:851. https://doi.org/10.1007/s10586-025-05520-9
4. Ziv J, Lempel A (1977) A universal algorithm for sequential data compression. IEEE Trans Inf Theory 23(3):337–343. https://doi.org/10.1109/TIT.1977.1055714
5. Cleary JG, Witten IH (1984) Data compression using adaptive coding and partial string matching. IEEE Trans Commun 32(4):396–402. https://doi.org/10.1109/TCOM.1984.1096090
6. Katz S (1987) Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Trans Acoust Speech Signal Process 35(3):400–401. https://doi.org/10.1109/TASSP.1987.1165125
7. Jayasankar U, Thirumal V, Ponnurangam D (2021) A survey on data compression techniques: from the perspective of data quality, coding schemes, data type and applications. J King Saud Univ Comput Inf Sci 33(2):119–140
8. Kwon B, Gong M, Huh J, Lee S (2018) Identification and restoration of LZ77 compressed data using a machine learning approach. In: 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 1787–1790 https://doi.org/10.23919/APSIPA.2018.8659755
9. Kempa D, Langmead B (2021) Fast and space-efficient construction of AVL grammars from the LZ77 parsing. In: 29th Annual European Symposium on Algorithms (ESA 2021). LIPIcs, vol. 204, p. 56:1–56:14 https://doi.org/10.4230/LIPIcs.ESA.2021.56
10. Weinberger MJ, Seroussi G, Sapiro G (2000) The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS. IEEE Trans Image Process 9(8):1309–1324. https://doi.org/10.1109/83.855427
11. Skodras A, Christopoulos C, Ebrahimi T (2001) The JPEG 2000 still image compression standard. IEEE Signal Process Mag 18(5):36–58. https://doi.org/10.1109/79.952804
12. Ballé J, Minnen D, Singh S, Hwang SJ, Johnston N (2018) Variational image compression with a scale hyperprior. arXiv preprint arXiv. https://doi.org/10.48550/arXiv.1802.01436. :1802.01436
13. Minnen D, Ballé J, Toderici GD (2018) Joint autoregressive and hierarchical priors for learned image compression. Adv Neural Inf Process Syst 31:10771–10780
14. Cheng Z, Sun H, Takeuchi M, Katto J (2020) Learned image compression with discretized Gaussian mixture likelihoods and attention modules. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7939–7948 https://doi.org/10.1109/CVPR42600.2020.00796
15. Castiglione A, Pizzolante R, De Santis A, Carpentieri B, Castiglione A, Palmieri F (2015) Cloud-based adaptive compression and secure management services for 3D healthcare data. Future Gener

Comput Syst 43–44:120–134
16. Bai T, Pan C, Han C, Hanzo L (2021) Reconfigurable intelligent surface aided mobile edge computing. IEEE Wirel Commun 28(6):80–86
17. Uthayakumar J, Vengattaraman T, Arjunan S (2022) An efficient near lossless image compression algorithm using dissemination of spatial correlation for remote sensing color images. Wirel Pers Commun 122(4):2963–2994
18. Boafo V (2026) Enhanced adaptive discriminant compression scheme to optimize data compression on mobile cloud systems. PhD thesis, School of Computing and Information Sciences, University of Technology and Applied Sciences, Ghana
19. Mairal J, Bach F, Ponce J, Sapiro G, Zisserman A (2009) Supervised dictionary learning. Adv Neural Inf Process Syst 21:1033–1040
20. Liang X, Zhao K, Di S, Li S, Underwood R, Gok AM, Tian J, Deng J, Calhoun JC, Tao D, Chen Z, Cappello F (2023) SZ3: a modular framework for composing prediction-based error-bounded lossy compressors. IEEE Trans Big Data 9(2):485–498
21. Özyılkan E, Erkip E (2024) Distributed compression in the era of machine learning: a review of recent advances. In: 58th Annual Conference on Information Sciences and Systems (CISS), pp. 1–6. IEEE
22. Li S, Lu J, Hu Y, Mattos LS, Li Z (2025) Towards scalable medical image compression using hybrid model analysis. J Big Data 12:45. https://doi.org/10.1186/s40537-025-01073-1
23. Moradi A, Rezaei F (2024) Intelligent and efficient task caching for mobile edge computing. Cluster Comput 27(10):14095–14112. https://doi.org/10.1007/s10586-024-04658-2
24. Sahkhar L, Balabantaray BK, Panda SK (2024) Mobility-enhanced delay-aware cloudlet movement and placement using cluster-based technique in a smart healthcare platform. Cluster Comput 27:11803–11821. https://doi.org/10.1007/s10586-024-04569-2
25. Bolte J, Sabach S, Teboulle M (2014) Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Math Program 146(1–2):459–494. https://doi.org/10.1007/s10107-013-0701-9
26. Maas W, Lorenzon A (2025) Exploring cloud instance options for optimal performance-cost efficiency. Cluster Comput 28:984. https://doi.org/10.1007/s10586-025-05702-5

Figures

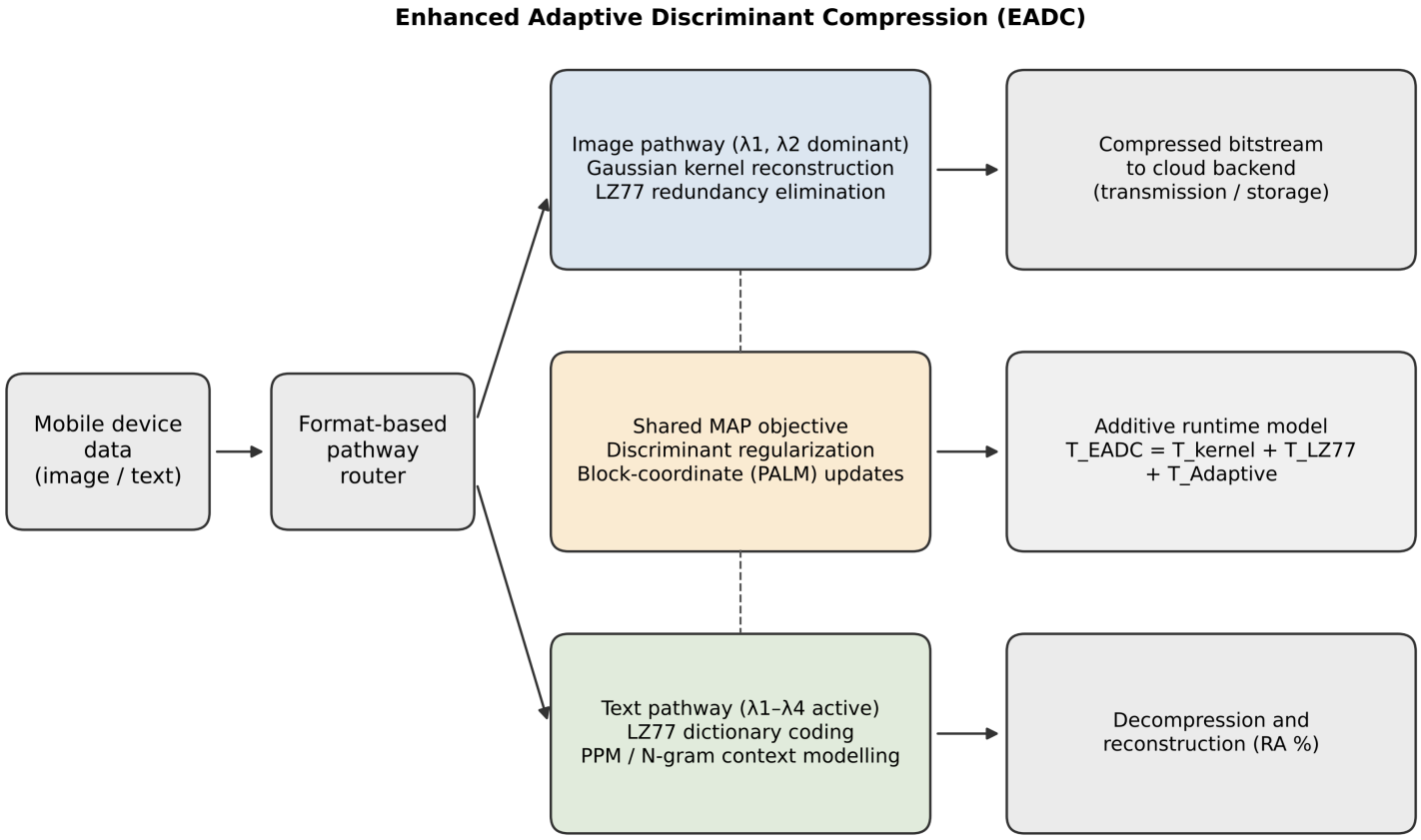


Figure 1

Conceptual architecture of the EADC dual-pathway framework. Both pathways share a discriminant-regularized MAP objective; the additive runtime model attributes total processing time to kernel, dictionary and adaptive-update stages