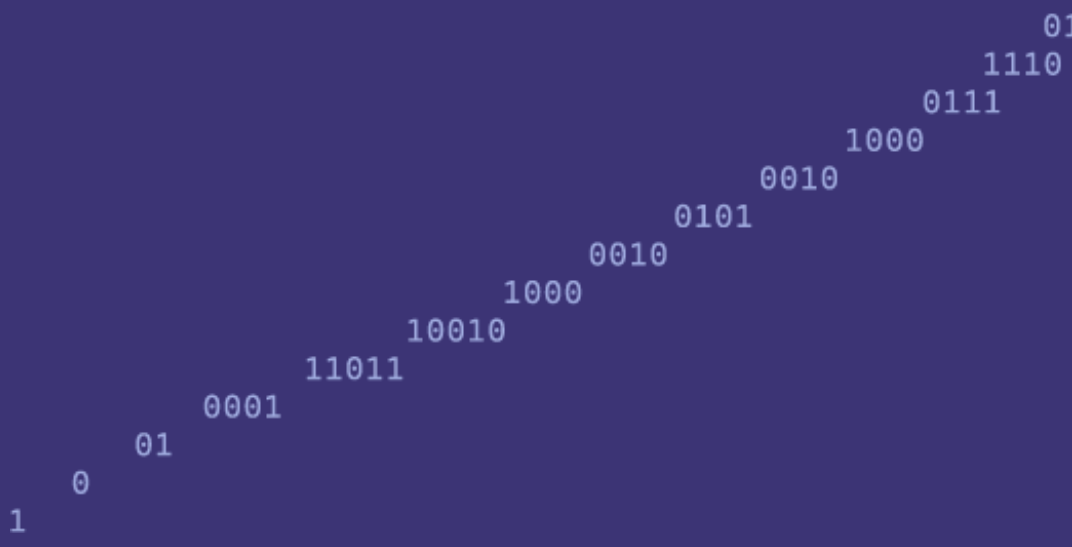
Datasent

WHITEPAPER · 2026



Datasent: An Overview

A lossless encoding layer for storage, bandwidth, and AI-native compute.

AUTHORED BY
Datasent Research

CONTACT
info@datasent.com · datasent.com

EDITION
November 2026

CONTENTS

Table of contents

A unified overview of the Datasent framework for a mixed technical and business audience. Full proofs, algorithms, and implementation details are deferred to the companion papers cited throughout.

ABSTRACT

Datasent: An Overview

A Lossless Encoding Layer for Storage, Bandwidth, and AI-Native Compute. Datasent Research · info@datasent.com · datasent.com · 2026.

Companion overview to the Datasent research series. This paper summarises the framework, its principal mathematical results, and its consequences for storage, bandwidth, machine learning, and large language model workloads. Full proofs, algorithms, and implementation details are deferred to the companion papers. One or more aspects of the technology described in this paper are the subject of pending patent applications.

ABSTRACT

Datasent is a lossless data encoding layer that occupies a position no prior representation occupies: it is at the same time compact, exactly invertible, structurally explicit, and computable on the encoded form. The encoded representation is produced once and used directly across storage, transport, analytics, and machine learning, without an intervening decode step for a broad class of operators. This paper provides a unified overview for a mixed technical and business audience. We state the principal mathematical results, summarise the systems and machine learning consequences, and map the supporting technical paper series. Proofs, algorithms, and reference implementations are given in the companion papers cited throughout.

SECTION 1

1 The data problem

Across industries, data is being generated faster than infrastructure can absorb it. Sensor and telemetry pipelines emit continuous streams at high resolution. Camera networks produce tens of gigabytes per hour. Transactional and behavioural systems record events at fine granularity. Scientific and industrial instruments routinely exceed the throughput of conventional ingestion pipelines.

Three constraints compound in this setting.

Storage. Total cost of maintaining large datasets grows with replication factors, backup obligations, and retention requirements, not only with raw volume.

Bandwidth. Moving data is frequently more expensive than storing it. Edge-to-cloud, cross-region, and inter-stage transfers stall pipelines when network capacity does not keep pace with generation.

Computation. A typical pipeline ingests, cleans, transforms, aggregates, and extracts features as separate stages. Each stage rereads, rewrites, or re-materialises large volumes of data, often reconstructing the same signal repeatedly.

Existing approaches treat these constraints in isolation. Classical compression reduces storage but produces an opaque byte stream that must be fully decompressed before use. Columnar storage preserves schema but does not exploit the signal structure within columns. Machine learning preprocessing extracts features tuned to a task but is typically lossy and runs outside the storage and transport layers.

The common pattern across these solutions is repeated conversion between incompatible forms,

$$\text{raw} \to \text{compressed} \to \text{decompressed} \to \text{processed}, \tag{1}$$

with each transformation incurring redundant cost and discarding information the next stage will need.

SECTION 2

2 The Datasent decomposition

The framework rests on a single observation about real-world data: within sufficiently small regions, structured signals exhibit local regularity that can be captured by a small number of parameters. The framework formalises this as a local decomposition,

$$x = f_\theta(\text{position}) + r, \tag{2}$$

in which a structured component f_θ captures the predictable part of a data region and a correction r records the remaining discrepancy. The two components have fundamentally different roles. The structured component is compact, interpretable, and directly usable. The correction is small in magnitude, low in entropy, and what makes the representation exactly invertible.

In the primary setting we work with a canonical integer matrix

$$Y \in \mathbb{Z}^{N \times D}, \tag{3}$$

where N is the number of rows (samples, timesteps, or ordered units) and D is the number of columns (features or channels). Integer representation is a deliberate design choice: it eliminates floating-point ambiguity across encoding and decoding environments, which is essential for exact reconstruction across heterogeneous systems.

The matrix is partitioned into S contiguous segments along its first axis,

$$Y = \begin{pmatrix} Y^{(1)} \\ Y^{(2)} \\ \vdots \\ Y^{(S)} \end{pmatrix}, \qquad Y^{(s)} \in \mathbb{Z}^{L_s \times D}, \qquad \sum_{s=1}^{S} L_s = N, \tag{4}$$

and each segment is encoded independently. For segment s under a chosen model family $\tau \in \mathcal{M}$, the framework produces a structured prediction $P^{(s)}$ and an exact integer correction $R^{(s)}$ such that

$$Y^{(s)} = P^{(s)} + R^{(s)}, \qquad P^{(s)} \in \mathbb{Z}^{L_s \times D}, \qquad R^{(s)} \in \mathbb{Z}^{L_s \times D}. \tag{5}$$

This is the framework's central identity. Every result that follows rests on it.

SECTION 3

3 The encoded representation

Each region of the dataset is represented as a structured token,

$$z_s = (\tau_s, C^{(s)}, R^{(s)}, m_s), \tag{6}$$

where $\tau_s \in \mathcal{M}$ identifies the model family, $C^{(s)} \in \mathbb{R}^{K_s \times D}$ is the compact parameter description of the structured component, $R^{(s)} \in \mathbb{Z}^{L_s \times D}$ is the exact integer correction, and m_s records segment length, dimensionality, and configuration. The encoded dataset is the ordered sequence

$$\mathcal{E}(Y) = (z_1, z_2, \ldots, z_S). \tag{7}$$

This token form is the unit of analysis throughout. Storage systems see $\mathcal{E}(Y)$ as the dataset. Network transports move tokens, not raw data. Analytical operators and machine learning models read τ_s, $C^{(s)}$, and the relevant statistics of $R^{(s)}$ directly from each token. When a downstream component requires the raw signal, decoding reproduces Y exactly.

A notable consequence, developed in the Trusted Setup paper, is that $C^{(s)}$ and $R^{(s)}$ play very different transmission roles. The correction $R^{(s)}$ is low entropy and is what makes reconstruction exact. The structured component $C^{(s)}$ describes a class of segments. The framework therefore admits an operating mode in which only $R^{(s)}$ and m_s cross the primary network boundary while $C^{(s)}$ is held on a separate channel under explicit authorisation. Raw data does not leave the source environment in any configuration.

SECTION 4

4 Four defining properties

The encoded representation has four properties that distinguish it, in combination, from any compression format or storage system known to the authors. Each property has precedent in isolation. The contribution is that all four hold simultaneously, under a single optimisation objective, and inside a single representation that persists through the data lifecycle.

Compactness. When local structure is strong, $(C^{(s)}, R^{(s)}, m_s)$ together require fewer bits than $Y^{(s)}$, reducing storage and transmission cost.

Losslessness. Y is always exactly recoverable from $\mathcal{E}(Y)$. Strict losslessness here means bitwise reconstruction of the canonical integer matrix, not bounded error.

Structure. The representation explicitly carries the local behaviour of the signal through τ_s and $C^{(s)}$, making this structure available to downstream systems without a separate extraction step.

Computability. A broad class of operators evaluates directly on $\mathcal{E}(Y)$ and returns the same result as decode-then-compute, at asymptotic cost that scales with the encoded size rather than the raw size.

SECTION 5

5 Principal results

5.1 Lossless reconstruction

> **THEOREM · LOSSLESS RECONSTRUCTION** (Whitepaper, Theorem 3.1)
>
> *Under the determinism conditions assumed by the framework and exact storage of $R^{(s)}$, for every segment s,*
>
> $$\mathcal{D}(z_s) = Y^{(s)}, \tag{8}$$
>
> *and consequently $\mathcal{D}(\mathcal{E}(Y)) = Y$.*

The encoding and decoding functions $\mathcal{E}$ and $\mathcal{D}$ therefore satisfy $\mathcal{D} \circ \mathcal{E} = \mathrm{id}$ on all canonical matrices. This is the foundation on which every other result rests. The proof and the precise determinism conditions are given in the foundational paper.

5.2 Compression through structure

Losslessness alone does not imply compression. Let $L(\cdot)$ denote the number of bits required to encode a quantity under a fixed lossless coding scheme. Compression for segment s occurs when

$$L_{\mathrm{tok}}(Y^{(s)}, \tau) < L_{\mathrm{raw}}(Y^{(s)}), \tag{9}$$

which holds when three conditions are jointly satisfied: the model captures most of the local structure, so $R^{(s)}$ has substantially lower entropy than $Y^{(s)}$; the parameter description $C^{(s)}$ is compact relative to the residual savings; and the per-token overhead is small relative to the total saving.

When these conditions hold, the framework achieves compression not by treating data as an opaque symbol stream but by modelling its structure and encoding only what the model fails to explain. Classical entropy coding may additionally be applied to $C^{(s)}$ and $R^{(s)}$ as a lower-level layer, potentially compounding the savings.

5.3 Adaptive selection

Because compression depends on the choice of model, the framework selects models adaptively. For each segment s, the optimal model family is

$$\tau_s^\star = \arg\min_{\tau \in \mathcal{M}} L_{\text{tok}}(Y^{(s)}, \tau). \tag{10}$$

This is a direct application of the Minimum Description Length principle: the selected model yields the most compact complete description of the data. The objective naturally balances model complexity against correction size. A model too simple leaves a large correction; a model too complex incurs high parameter cost. The optimum sits where marginal gains in correction compression are exactly offset by the marginal cost of additional parameters.

5.4 Adaptive segmentation

Segment boundaries are not fixed. The framework jointly optimises segmentation and model assignment over the whole dataset. Let $V(t)$ denote the minimum total description length for $Y_{1:t,:}$. Adaptive segmentation admits optimal substructure:

$$V(t) = \min_{1 \le s \le t} \{V(s-1) + C_\lambda(s,t)\}, \qquad V(0) = 0, \tag{11}$$

where $C_\lambda(s,t)$ denotes the minimum description length over models in $\mathcal{M}$ for the candidate segment $Y_{s:t,:}$ plus a per-segment penalty $\lambda \ge 0$. The penalty λ provides monotone, interpretable regularisation: $|\mathcal{S}^\star(\lambda)|$, the number of segments at optimum, is non-increasing in λ.

> **THEOREM · OPTIMAL SUBSTRUCTURE** (Whitepaper, Theorem 7.1)
>
> *The dynamic programming recurrence on $V(\cdot)$ computes the minimum total description length over all valid segmentations and model assignments for the full dataset Y.*

The algorithmic details, including the windowed and greedy variants, are given in the foundational paper.

5.5 Operator compatibility

The core computational result of the framework rests on a single decomposition. For any linear operator $\mathcal{A}$,

$$\mathcal{A}(Y^{(s)}) = \mathcal{A}(P^{(s)}) + \mathcal{A}(R^{(s)}). \tag{12}$$

THEOREM · LINEAR OPERATOR COMPATIBILITY (Whitepaper, Theorem 8.3)

If both terms on the right are computable from z_s without reconstructing $Y^{(s)}$, then $\mathcal{A}$ is encoding-compatible: there exists $\tilde{\mathcal{A}}$ acting on tokens such that $\mathcal{A}(Y) = \tilde{\mathcal{A}}(\mathcal{E}(Y))$.

The condition is satisfied by column-wise aggregation, linear filtering and convolution, inner products with fixed vectors, finite differences, rolling statistics, standardisation, feature scaling, and linear projection. Order statistics (median, quantiles, top-k), rank-dependent operators, and non-linear pointwise transformations are not encoding-compatible without partial reconstruction; they fall back to decode-on-demand on the affected segments, which is exact by Theorem 3.1.

5.6 Token-space speedup

THEOREM · TOKEN-SPACE SPEEDUP (Whitepaper, Theorem 8.9)

Let $\bar{\rho}$ be the average correction density. The speedup of token-space computation over decode-then-compute is

$$\Theta\left(\frac{N}{SK_{\max} + \bar{\rho}N}\right). \tag{13}$$

As $\bar{\rho} \to 0$ and $S \ll N/K_{\max}$, this grows as $\Theta(L_{\text{avg}}/K_{\max})$,

where $L_{\text{avg}} = N/S$ is the average segment length and $K_{\max}$ is the maximum parameter count per segment. The ratio $L_{\text{avg}}/K_{\max}$ is the compression ratio in operator form. For a sensor stream with $L_{\text{avg}} = 10^3$ and $K_{\max} = 4$, token-space aggregation is roughly two orders of magnitude faster than decode-then-aggregate.

SECTION 6

6 Complexity

The framework has linear asymptotic encoding and decoding costs.

OPERATION	COMPLEXITY
Encoding	$O(N \cdot \lvert\mathcal{M}\rvert \cdot K_{\max}(K_{\max} + D))$
Decoding	$O(NK_{\max}D)$
Adaptive segmentation (windowed)	$O(NW\lvert\mathcal{M}\rvert K_{\max}^2 D)$
Token-space linear operator	$O(SK_{\max}D + \bar{\rho}ND)$
Streaming (per row)	$O(K_{\max}^2 D\lvert\mathcal{M}\rvert)$

For fixed $|\mathcal{M}|$, $K_{\max}$, and D, encoding and decoding are $O(N)$. Decoding is faster than encoding by a factor of $O(|\mathcal{M}|K_{\max})$. Segmentation is parameterised by a maximum segment window W; the foundational paper develops the optimality and complexity trade-off. Token-space operators scale with the number of segments S and the correction density $\bar{\rho}$, not with N.

SECTION 7

7 Bandwidth and storage

Define b as the bits per entry of the raw matrix, $\bar{b}_R$ as the average bits per entry of the correction, and $M_{\text{raw}} = NDb$ as the total raw size in bits. The total encoded size is

$$M_{\text{tok}} = O\Big(SK_{\max}D + ND\bar{b}_R\Big), \tag{14}$$

and the ratio of encoded size to raw size is

$$\frac{M_{\text{tok}}}{M_{\text{raw}}} = \frac{SK_{\max}D + ND\bar{b}_R}{NDb} = O\left(\frac{K_{\max}}{L_{\text{avg}}}\right) \quad \text{when} \quad \bar{b}_R \ll b. \tag{15}$$

Compression holds for sufficiently large N whenever $\bar{b}_R < b$. The same ratio governs storage reduction, bandwidth reduction on edge-to-cloud and inter-stage transfers, and replication cost. Under the trusted-setup transmission protocol developed in the Trusted Setup paper, only $R^{(s)}$ and m_s cross the primary network boundary, so the bandwidth reduction is at the rate $\bar{b}_R/b$ in the low-correction setting.

SECTION 8

8 Multi-modal coverage

The framework is general with respect to modality. Every supported modality decomposes into one or more canonical components, each handled by the same primitive.

The principal canonical form is the integer matrix, which covers tabular data, sensor streams, and time series. Imaging data is handled by treating frames or tiles as matrices over pixel grids. Video adds a motion-residual decomposition, developed in the Temporal Tokens paper, that generalises classical motion compensation into a token-native form. Audio is handled either as a waveform matrix or in a spectral representation. Text and embeddings are handled as matrices over vocabulary or embedding dimensions. Graphs are handled as adjacency and feature matrices.

In each case the lossless reconstruction guarantee, the adaptive selection rule, and the operator-compatibility property carry over without modification. The same encoding layer serves modalities that are typically handled by separate, specialised infrastructure.

SECTION 9

9 Machine learning integration

The encoded representation has a property conventional compression formats do not. The structured component of each token is itself a feature vector. The same representation that serves storage and transport is therefore also a feature representation, available to downstream models without a separate preprocessing pass.

9.1 Three operating modes

Mode 1 (raw-equivalent). The framework decodes a region on demand and presents the original signal to the model. Decoding is exact by Theorem 3.1, so no model code is modified.

Mode 2 (coefficient features). The model consumes $C^{(s)}$ directly, optionally augmented with a small summary statistic of $R^{(s)}$ such as residual variance. This is the most compact and most efficient mode at training time.

Mode 3 (hybrid). The model consumes $C^{(s)}$ together with a quantised projection of $R^{(s)}$. This trades a controlled increase in feature dimension for an explicit correction-side signal.

9.2 Training fidelity

The Training Fidelity paper proves two results that bound the loss gap between training on raw data and training on token features.

Zero-gap. When the task label Y is a deterministic function of $C^{(s)}$, training in Mode 2 incurs zero loss gap relative to training on the raw signal. The entire task-relevant signal lives in the coefficient subspace; the correction carries no information the model could use.

Bounded-gap. When the task label is not fully captured by $C^{(s)}$, the loss gap in Mode 2 satisfies

$$\left| \mathcal{L}(f_\theta^{\mathrm{raw}}) - \mathcal{L}(f_\theta^{\mathrm{tok}}) \right| \;\le\; 2L_\ell\, L_f \sqrt{D}\, \sigma_X \sqrt{\alpha(K)}, \tag{16}$$

where L_ℓ is the Lipschitz constant of the loss, L_f is the Lipschitz constant of the model, D is the input dimensionality, σ_X is the input signal scale, and $\alpha(K) = \sigma_R^2/\sigma_X^2$ is the correction-to-signal variance ratio at model order K. The bound is small whenever the structured component captures most of the signal variance.

Hybrid quantisation. Under Mode 3 with b-bit correction quantisation, the correction-to-signal ratio in the bound is replaced by

$$\alpha_b(K) \;\le\; \frac{\sqrt{3\pi/2}}{4^b}\,\alpha(K), \tag{17}$$

which decays exponentially in b. The proportionality constant comes from a result in the broader machine learning systems literature, TurboQuant (arXiv:2504.19874), which establishes the inner-product error bound for randomised rotation plus uniform quantisation.

9.3 Strategy selection diagnostic

A single diagnostic determines which mode is appropriate. Regress the task label on $\mathrm{vec}(C^{(s)})$ using ordinary least squares and compute

$$R_K^2 \;=\; 1 \;-\; \frac{\mathrm{Var}(Y - \hat{Y}_K)}{\mathrm{Var}(Y)}. \tag{18}$$

The value of R_K^2 determines the operating mode.

R_K^2	STRATEGY	APPROXIMATE COMPRESSION
≥ 0.95	Mode 2 (coefficients)	**Full compression ratio**
0.85 **to** 0.95	Mode 2 plus correction summary	**Full compression ratio**
0.70 **to** 0.85	Mode 3, $b = 2$ to 4	$3\times$ **to** $10\times$
< 0.70	Mode 3, $b \geq 8$, or Mode 1 fallback	$1\times$ **to** $4\times$

The diagnostic is computed once per dataset and per task.

9.4 Computational saving

The Training Fidelity paper establishes that token-based training reduces per-step cost by a factor of $L/K = \mathrm{CR}$ for input-dimension-bound models, and by L^2/K^2 for attention-based models whose cost scales quadratically in sequence length. Combined with reduced I/O, batch memory, and amortised preprocessing, the total computational saving over the training run is

$$\frac{T_{\text{raw}}^{\text{total}}}{T_{\text{tok}}^{\text{total}}} \geq \text{CR}, \tag{19}$$

with equality for I/O-bound training and strict inequality for compute-bound transformers.

SECTION 10

10 Large language models

The same encoded representation supports four large language model use cases without modification.

Fine-tuning. Token-coefficient features feed fine-tuning runs directly. The bounded-gap result applies because fine-tuning is supervised training. The Lipschitz constants and model class are identical to the pretraining setting.

Key-value cache compression. The framework integrates with attention KV caches. For TurboQuant b -bit reconstruction of keys $\tilde{K}$ from K, the per-query inner-product error is bounded by

$$\mathbb{E}\Big[\,\big|\langle q, k_t\rangle - \langle q, \tilde{k}_t\rangle\big|^2\,\Big] \;\leq\; \frac{\sqrt{3\pi/2}}{d_k \cdot 4^b}, \qquad \forall\, t \in [n], \tag{20}$$

for unit-norm queries $q \in \mathbb{R}^{d_k}$. The attention-weight ℓ_1 error propagates through the softmax as

$$\big\|\text{softmax}(Q\tilde{K}^\top/\sqrt{d_k}) - \text{softmax}(QK^\top/\sqrt{d_k})\big\|_1 \;\leq\; n \cdot \frac{\sqrt{3\pi/2}}{4^b\sqrt{d_k}}. \tag{21}$$

The LLM Integration paper develops explicit bit-width settings. Quality-neutral inference is achievable at $b = 3.5$ bits (roughly 4.5x compression of the cache); marginally lossy inference at $b = 2.5$ bits (roughly 6.4x).

Retrieval-augmented generation. Embeddings are tokenised at the same level of compression as other dense modalities. A bound establishes the conditions under which nearest-neighbour ranking is preserved under q-bit quantisation: if

$$q \;\geq\; \left\lceil \log_2\!\left(1 + \frac{2s}{\Delta^\star}\right)\right\rceil, \qquad \Delta^\star = \frac{\gamma_{\min}}{2\sqrt{D}}, \tag{22}$$

then the top-ranked retrieval result is preserved for every query whose margin to the second-best result is at least $\gamma_{\min}$. Typical operating points compress embeddings by two to three orders of magnitude.

Agentic memory and protocol integration. Long-running agents accumulate session memory at a rate that conventional context windows cannot absorb. The framework compresses memory at the encoding layer and exposes it through a small set of tools usable by agents that follow the Model Context Protocol. Retrieval against compressed memory uses the same nearest-neighbour bound as retrieval-augmented generation.

The LLM Integration paper develops each use case in full, including the three transformer integration points (embedding level, key-value cache level, and context level) and multi-modal context compression.

SECTION 11

11 Systems implications

The framework has direct consequences across the data stack. We summarise without re-deriving.

Storage. Primary storage footprint is reduced by the compression ratio $L_{\text{avg}}/K_{\max}$. Replication and backup costs scale by the same factor.

Bandwidth. Total data crossing the network is reduced by the same ratio. Under the trusted-setup transmission protocol, only the correction $R^{(s)}$ and metadata m_s cross the primary network. Raw data does not leave its source environment.

Pipeline simplification. For a pipeline of K encoding-compatible operators, total materialised data is $O(K \cdot M_{\text{tok}})$ versus $O(K \cdot M_{\text{raw}})$ under the conventional model, a reduction by the compression ratio at every stage.

Distributed and streaming systems. Segments are independent, so the framework offers perfect parallelism, segment-local fault isolation, append-only updates, and adaptive rate control by adjusting the segment penalty λ.

Positioning in the stack

LAYER	REPRESENTATIVE SYSTEMS
Applications	Analytics, ML models, dashboards
Feature pipelines	Preprocessing, normalisation, aggregation
Encoding layer	**Datasent structured tokenisation**
Storage / transport	Object storage, message queues, databases
Hardware	Disks, network interfaces, memory

The encoding layer is deployable on standard infrastructure. Current deployments include a state transportation agency operating a real-time camera-to-AI pipeline at sub-two-second end-to-end latency, in which roadside cameras encode video into compact event tokens at the edge and only tokens cross the network to the analytics platform. A second deployment pattern serves analytics

against sensitive data held inside a customer environment, where the encoder runs alongside the data, encodes it in place, and only tokens leave the environment for analytics, model training, and API scoring.

SECTION 12

12 Relation to existing work

The framework sits at the intersection of several established research areas. Several prior systems exhibit one or two of the four defining properties; the contribution is the combination.

Classical lossless compression (entropy coders, dictionary coders). Lossless but opaque. The framework's correction $R^{(s)}$ can be passed to an entropy coder as a lower layer, potentially compounding savings.

Transform coding (DCT, wavelet). Lossy or near-lossless depending on quantisation. Exposes a frequency-domain structure but does not produce a strict-lossless, computable representation under a single adaptive selection objective.

Columnar storage formats (Parquet, ORC). Preserve schema and support column-level pushdown. Run-length and delta encoding are special cases of degree-zero and degree-one models in $\mathcal{M}$ but applied as fixed heuristics rather than adaptive selections.

Time-series segment-and-model systems (ModelarDB and related). The closest precedent. Adaptively fit per-segment models and support aggregation queries directly on parameters. Differ from the framework in that they do not provide strict-lossless reconstruction with exact integer correction storage, do not jointly optimise selection and segmentation under a single MDL objective, and do not extend across multi-modal data with a single primitive.

Foundation-model time-series tokenisers (Chronos, TimesFM). Tokenise time series for downstream model consumption. Lossy by design. The framework's tokenisation is exact, and the token coefficients are themselves usable as model features without a learned tokenisation step.

Federated learning. Keeps raw training data local. Lossy in aggregation and application-specific. The framework provides lossless exchange of corrections across any modality, with explicit authorisation for reconstruction.

Secure data clean rooms. Provide governed access to shared data but typically require raw data to enter a controlled environment. The framework eliminates this requirement: raw data does not leave the source environment in any configuration.

A more detailed comparison spanning thirteen prior-art system classes is given in the foundational paper.

SECTION 13

13 Verifiability

The decoding computation is arithmetisable. This means that a zero-knowledge proof system can produce a succinct proof that a given encoded representation decodes to a value satisfying a stated property, without revealing the value itself. Formally, decoding admits an arithmetic-circuit representation of size

$$O(L_s \cdot K_s \cdot D) \tag{23}$$

per segment. The framework does not depend on this integration; the four defining properties hold without it. For applications where a party needs to demonstrate compliance with a policy without disclosing the underlying data, the framework provides a natural substrate for verifiable computation. The foundational paper develops this in a dedicated section.

SECTION 14

14 Paper series

This overview summarises a series of technical papers, each of which develops one component of the framework in full.

1. **A Mathematical Framework for Structured Information Encoding.** The foundational paper. Defines the framework, proves Theorem 3.1, formalises the MDL objective, develops adaptive segmentation, and establishes the token algebra and operator-compatibility theorem.
2. **A Trusted Setup for Bandwidth-Minimal, Lossless Data Tokenisation.** The transmission and governance paper. Develops the protocol under which only $R^{(s)}$ and m_s cross the primary network, and proves the security of the structure-correction separation.
3. **ML Feature Tokens.** Develops the use of token coefficients as machine learning features and the integration with standard preprocessing pipelines.
4. **Training Fidelity.** Proves the zero-gap and bounded-gap training results, develops the R^2_K diagnostic, and establishes the computational-saving theorems for token-based training.
5. **Pseudocode Reference.** Implementation patterns for the encoding, decoding, segmentation, model selection, diagnostic, and operator-evaluation procedures.
6. **LLM Integration.** Extends the framework to transformer fine-tuning, key-value cache compression, retrieval-augmented generation, and agentic memory.
7. **FABR Empirical.** Reports compression-ratio, encoding-throughput, and token-space speedup measurements across standard benchmarks and representative real-world workloads.

8. **Meta-Compressor.** Signal-adaptive mechanism selection, in which a diagnostic stage routes each segment to the appropriate compression mechanism rather than committing to a single family in advance.
9. **Compression Proof.** Formal compression-ratio guarantees across time-series, video, and image modalities, with explicit accounting for protocol overhead.
0. **Temporal Tokens.** Motion-residual decomposition for video and spatiotemporal data.
1. **Structural Reuse in FABR.** Codebook learning and description-length bounds for recurring motifs.

SECTION 15

15 Closing

The framework defines a new layer of the data stack. It is positioned between storage and feature pipelines and is designed to remain in encoded form across the full data lifecycle. Compression, structured representation, and computation are not separate concerns within this layer but components of a single optimisation, the MDL objective over the model family $\mathcal{M}$ and the segmentation $\mathcal{S}$.

The central pipeline of the framework is

$$Y \xrightarrow{\mathcal{E}} \mathcal{E}(Y) \xrightarrow{\tilde{\mathcal{A}}} \mathcal{A}(Y) \xrightarrow{\pi} \text{verified claim}, \tag{24}$$

in which encoding, computation, and optionally verification operate in sequence without access to raw Y. Together these layers define a representation paradigm in which compression, computation, and verification are not separate concerns but components of a unified system.

The framework is fully implemented and operational. Deployments span a real-time camera-to-AI pipeline at a state transportation agency and in-environment analytics against sensitive customer data. The supporting research series develops each component in full mathematical detail. This overview is intended as the entry point.

For the full mathematical framework, see the foundational paper. For transmission, governance, machine learning, large language model, and empirical results, see the companion papers listed in Section 14. For commercial enquiries, contact ***info@datasent.com****.*