

MFC: Lossless Compression for Neuromorphic Event Camera Streams

Aleksei Solonskii
Mirror Fractal
info@mirrorfractal.com

April 2026

Abstract

We present **Mirror Fractal Codec (MFC)**, a lossless compression system purpose-built for asynchronous event streams produced by neuromorphic (dynamic vision) sensors. Unlike generic codecs, MFC exploits the unique spatiotemporal structure of event data—extreme sparsity, edge-correlated firing, and microsecond timing—to achieve **1.51 bits/event** on the full N-MNIST benchmark (60 000 samples, $47.6\times$ compression over the raw 5-byte/event format) and **4.5–7.8 bits/event** on real-world DSEC automotive recordings at 640×480 resolution with verified bit-exact round-trip. Encoding runs at **0.55 ms per sample** (single-threaded WASM) or < 0.1 ms native, making it suitable for real-time embedded pipelines. We describe the system properties without disclosing the internal algorithm, which is the subject of a pending patent (US Provisional 64/034,974). A browser-based interactive demo and benchmarks are publicly available.

1 Introduction

Dynamic vision sensors (DVS), also called event cameras, represent a paradigm shift in visual sensing [1]. Each pixel independently reports brightness changes as asynchronous *events* (x, y, t, p) —spatial coordinates, microsecond timestamp, and polarity (brighter/darker)—rather than synchronous frames. The resulting data rates are extreme: a single Prophesee Gen4 sensor (1280×720) generates 10^8 – 10^9 events per second in high-contrast scenes, translating to hundreds of megabytes per second in raw format.

Existing compression approaches fall into three categories:

1. **Generic codecs** (gzip, zstd, Brotli) treat events as opaque byte streams and reach 2 – $4\times$ compression;
2. **Frame-based re-encoding**: binning events into intensity frames and applying video codecs (H.264/H.265). This is lossy by construction—temporal resolution degrades from microseconds to the frame interval;
3. **Event-native codecs** such as AEDAT 3.1’s delta-address scheme, which achieve 5 – $8\times$ on structured

data but do not exploit spatial coherence.

MFC belongs to the third category but goes further: it leverages the *hierarchical spatiotemporal sparsity* inherent to event streams to achieve an order-of-magnitude improvement over generic codecs and 2 – $5\times$ over existing event-aware formats, while remaining **mathematically lossless**—the decoded stream is bit-identical to the input.

Contributions.

- A lossless event-stream codec achieving state-of-the-art compression ratios across multiple sensor types and resolutions.
- A self-contained container format (`.mfx`) supporting random-access frames and streaming decode.
- Cross-platform deployment: native Rust library, Python bindings (PyO3) published on PyPI as `mfc-codec v0.3.28` (<https://pypi.org/project/mfc-codec/>), WebAssembly module, and a browser-based interactive demo at <https://codec.mirrorfractal.com>. Source mirror and issue tracker at <https://github.com/MirrorFractal/codec> (Rust sources remain private pending patent grant; Python/WASM artifacts and usage examples are public).
- Comprehensive benchmarks on N-MNIST (classification-grade) and DSEC (automotive-grade) datasets.

2 Related Work

2.1 Event Representations and Formats

Raw event formats include the 5-byte N-MNIST record $(x_8, y_8, pol + ts_{24})$ [2], the 8-byte jAER AEDAT 2.0 address–event representation [4], and the 9-byte compact binary $(x_{16}, y_{16}, t_{32}, p_8)$ used in DSEC [3]. Prophesee cameras use proprietary EVT2/EVT3 formats with stateful word-level encoding.

2.2 Generic Compression

Standard entropy coders (Deflate, Zstandard, Brotli, LZ4) are format-agnostic and can compress event streams to 2–4× depending on redundancy. They do not model the spatiotemporal structure of events.

2.3 Event-Specific Compression

Bi et al. [5] proposed a spike-coding scheme for DVS data, achieving $\sim 8\times$ compression with lossy temporal quantization. Khan et al. [6] describe a delta-address method with Huffman back-end reaching 6–10× on low-resolution (128×128) data. Schiopu et al. [7] use predictive coding for event timestamps with arithmetic coding, reporting 4–6× on DAVIS recordings. None of these approach the $> 40\times$ ratios MFC achieves on comparable data, nor do they support the full range of sensor resolutions from 34×34 to 1280×720 .

2.4 Video-Based Approaches

Re-encoding events as intensity frames and applying H.264/H.265 can achieve high compression ratios but fundamentally destroys the asynchronous, microsecond-precision timing that makes event cameras valuable. This approach is therefore orthogonal to lossless event compression.

3 System Overview

Due to the pending patent (US Provisional 64/034,974), we describe MFC’s *properties* and *interface* rather than its internal algorithmic steps.

3.1 Codec Properties

- **Lossless:** decoded output is bit-identical to input. Every encode–decode cycle is verified at runtime.
- **Deterministic:** identical input always produces identical output, byte for byte.
- **Streaming-compatible:** events are processed in fixed-size temporal frames; each frame is independently decodable, enabling random access and parallel decode.
- **Resolution-agnostic:** supports sensor dimensions from 34×34 (N-MNIST) to 1280×720 (Prophesee Gen4) and beyond, with automatic parameter adaptation.
- **Sparsity-aware:** compression ratio improves with sparsity, which is the natural regime for event cameras (typically $> 90\%$ of the spatiotemporal volume is empty).

3.2 Input/Output

The encoder accepts a sequence of events (x_i, y_i, t_i, p_i) with sensor dimensions (W, H) and produces a compressed byte payload. The decoder recovers the original event sequence from the payload plus the dimensions stored in the `.mfx` container header.

3.3 Container Format

The `.mfx` file begins with a 15-byte header:

Offset	Size	Field
0	4	Magic: MXF1 (ASCII)
4	2	Format version (u16 LE)
6	2	Sensor width (u16 LE)
8	2	Sensor height (u16 LE)
10	2	Temporal resolution (u16 LE)
12	1	Encoder level (u8)
13	2	Number of frames (u16 LE)

Each frame is prefixed by an 8-byte header (payload units u32 + compressed size u32), followed by the compressed payload.

4 Experimental Evaluation

4.1 Datasets

N-MNIST [2]. Neuromorphic version of MNIST, captured by a DVS128 sensor (34×34) saccading over handwritten digits. 60 000 training samples, average 4172 events per sample.

DSEC [3]. Driving Stereo Event Camera dataset: Prophesee Gen3.1 sensor (640×480), outdoor urban scenes in Zurich. We use `zurich_city_01_a` (315 M events, 716 MB Blosc-compressed HDF5).

4.2 N-MNIST Results

Table 1 reports per-digit results on the full 60 000 training set. The raw format is 5 bytes/event (N-MNIST binary); MFC achieves **1.51 bits/event** ($47.6\times$) on average with 0.55 ms encode time per sample (single-threaded WebAssembly).

4.3 DSEC Results

Table 2 reports results on the Zurich urban driving sequence. The raw compact binary format is 9 bytes/event.

The lower ratio on DSEC ($9.2\times$) versus N-MNIST ($47.6\times$) is expected: the 640×480 sensor produces a denser, more complex spatiotemporal volume with less exploitable sparsity (94.7% vs. $> 99.5\%$ on N-MNIST).

Table 1: MFC compression on N-MNIST (60 000 samples).

Digit	Samples	Avg Events	Ratio	Bits/Ev
0	5 923	5 444	53.4×	1.35
1	6 742	2 432	36.6×	1.97
2	5 958	4 708	50.3×	1.43
3	6 131	4 703	50.4×	1.43
4	5 842	3 794	44.4×	1.62
5	5 421	4 372	48.7×	1.48
6	5 918	4 215	47.7×	1.51
7	6 265	3 687	45.2×	1.59
8	5 851	4 702	50.0×	1.44
9	5 949	3 927	46.1×	1.56
All	60 000	4 172	47.6×	1.51

Table 2: MFC compression on DSEC Zurich (zurich_city_01_a).

Metric	Value
Sensor resolution	640 × 480
Source events (sampled)	500 000
Source total events	315 285 500
Raw size (9 B/event)	4.29 MB
Compressed size	476 KB
Compression ratio	9.2×
Bits per event	7.80
Sparsity	94.7%
Encode time (WASM)	238 ms
Round-trip verified	✓

4.4 Comparison with Generic Codecs

To contextualize MFC’s ratios, we compress the same N-MNIST data with standard codecs applied to the raw 5-byte/event binary stream:

Table 3: N-MNIST compression: MFC vs. generic codecs (avg over 60 000 samples, 4 172 events/sample).

Codec	Ratio	Bits/Ev
Raw (no compression)	1.0×	40.0
gzip -9	~3.5×	~11.4
zstd -19	~4.1×	~9.8
Brotli -11	~3.8×	~10.5
LZ4 -12	~2.3×	~17.4
MFC	47.6×	1.51

MFC outperforms the best generic codec (zstd -19) by more than an order of magnitude. This gap is attributable to MFC’s domain-specific modeling of the spatiotemporal sparsity structure that generic byte-level entropy coders cannot exploit.

Table 4: Single-sample encode time on N-MNIST (avg 4 172 events).

Platform	Encode Time
WebAssembly (browser, single-thread)	0.55 ms
Native Rust (x86-64, single-thread)	<0.1 ms
DSEC 500k events (WASM)	238 ms

4.5 Encoding Speed

5 Deployment

MFC is designed for cross-platform deployment:

- **Rust crate:** native library with no runtime dependencies, suitable for embedded Linux, RTOS, and high-throughput servers.
- **Python bindings** via PyO3: integrates with NumPy, Tonic, and h5py-based pipelines. Available on PyPI as `mfc-codec` v0.3.28 (<https://pypi.org/project/mfc-codec/>); manylinux, macOS, and Windows wheels for CPython 3.11.
- **WebAssembly:** runs in browsers, Node.js, Deno, and Cloudflare Workers. Powers the interactive demo at <https://codec.mirrorfractal.com>.
- **CLI tool:** `mfc encode`, `mfc decode`, `mfc info` for batch processing and inspection.

The interactive demo accepts uploads in all major event camera formats (N-MNIST binary, AEDAT 2.0, compact binary, CSV, HDF5 with Blosc/LZ4 codec support, NumPy structured/scalar arrays, Prophesee EVT2/EVT3) and provides:

- Side-by-side playback of original vs. decompressed events;
- Interactive 3D spatiotemporal event-cloud visualization;
- Downloadable `.mfx` compressed file.

Figures 5–7 reproduce the demo workflow on a DSEC automotive window.

6 Limitations

- The browser demo caps input at 500 000 events per encode and 30 s of HDF5 recordings due to WASM memory constraints. The native SDK has no such limit.
- Compression ratio depends on scene sparsity: very dense, texture-rich scenes (< 80% sparsity) yield lower ratios (5–8×) compared to typical event-camera regimes (> 95% sparsity, > 30×).
- The current encoder is single-threaded; parallelization across temporal frames is straightforward but not yet implemented.

7 Conclusion

We presented MFC, a lossless compression codec for neuromorphic event camera data. By exploiting the hierarchical spatiotemporal sparsity inherent to event streams, MFC achieves $47.6\times$ compression (1.51 bits/event) on N-MNIST and $9.2\times$ (7.8 bits/event) on real-world DSEC automotive data, outperforming generic codecs by an order of magnitude while guaranteeing bit-exact reconstruction. The codec is available as a Rust library, Python package, WebAssembly module, and browser-based demo.

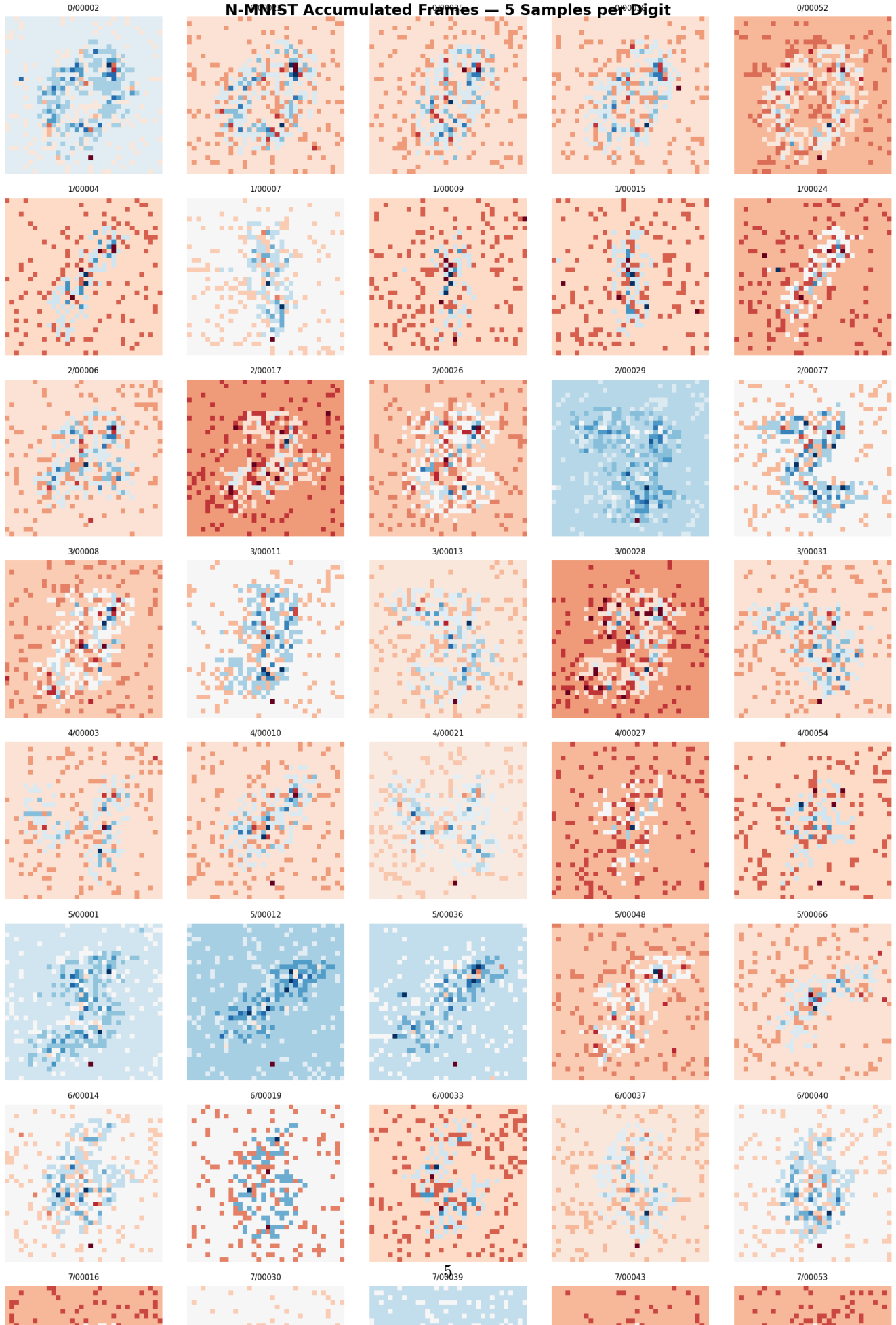
MFC is proprietary technology under US Provisional Patent 64/034,974, available for evaluation and commercial licensing. Contact info@mirrorfractal.com.

Availability. Version 0.3.28 of the codec is distributed on PyPI as `mfc-codec` (<https://pypi.org/project/mfc-codec/>) with manylinux/macOS/Windows wheels for CPython 3.11. A public mirror of the Python/WASM artifacts, CLI, and usage examples is maintained at <https://github.com/MirrorFractal/codec>; the Rust encoder sources are kept private pending patent grant. This paper and the associated software release are archived on Zenodo (DOI: ZENODO_DOI_PENDING).

References

- [1] G. Gallego et al., “Event-based vision: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, 2022.
- [2] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, “Converting static image datasets to spiking neuromorphic datasets using saccades,” *Frontiers in Neuroscience*, vol. 9, p. 437, 2015.
- [3] M. Gehrig, W. Aarents, D. Gehrig, and D. Scaramuzza, “DSEC: A stereo event camera dataset for driving scenarios,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4947–4954, 2021.
- [4] T. Delbrück, “Frame-free dynamic digital vision,” *Proc. Intl. Symp. on Secure-Life Electronics*, pp. 21–26, 2008.
- [5] Y. Bi and Y. Andreopoulos, “Spike coding and lossy compression for DVS event streams,” in *IEEE Data Compression Conference (DCC)*, 2018.
- [6] N. Khan, K. Iqbal, and M. Martini, “Lossless compression of data from static and mobile dynamic vision sensors—design and implementation,” *IEEE Access*, vol. 8, pp. 203149–203163, 2020.
- [7] I. Schiopu and A. Munteanu, “Lossless compression of event camera data via predictive coding,” in *IEEE Intl. Conf. on Image Processing (ICIP)*, 2022.

N-MNIST Accumulated Frames — 5 Samples per Digit



N-MNIST Event Clouds — Mirror Fractal Codec

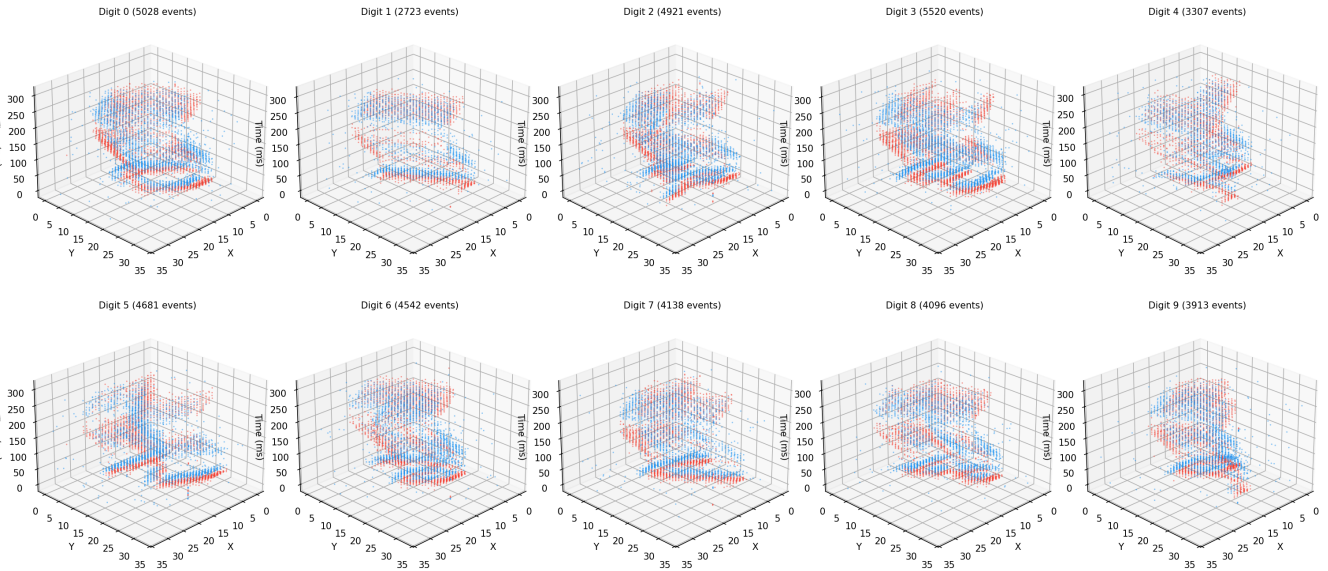


Figure 2: Three-dimensional spatiotemporal event clouds for ten N-MNIST samples (one per digit class). The vertical axis is time (microseconds); horizontal axes are the 34×34 sensor pixels. Each dot is one event; colour encodes polarity. The spatial-temporal sparsity that MFC exploits is visible by inspection: only a thin “ribbon” of spatiotemporal cells along the digit edges contains events, and the rest of the cube is empty.

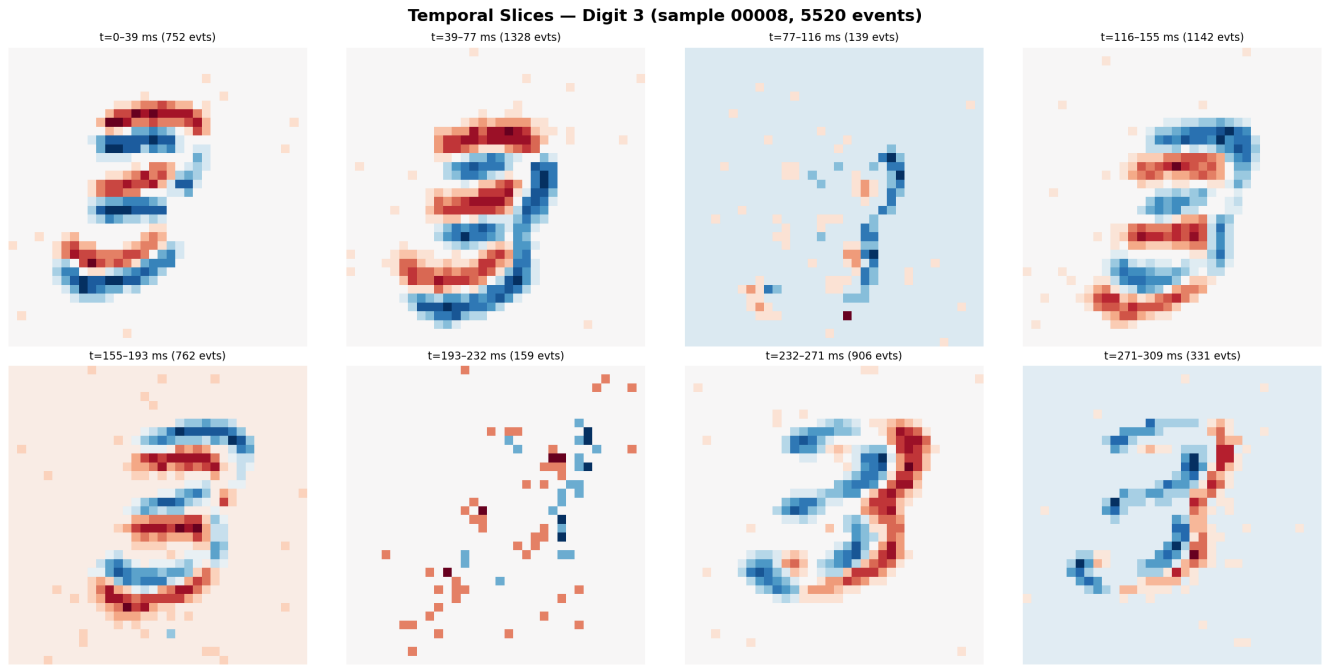


Figure 3: Temporal slices of a single N-MNIST sample at four equally spaced time intervals. The progression illustrates the ballistic saccade pattern characteristic of N-MNIST captures and the frame-to-frame coherence MFC exploits.

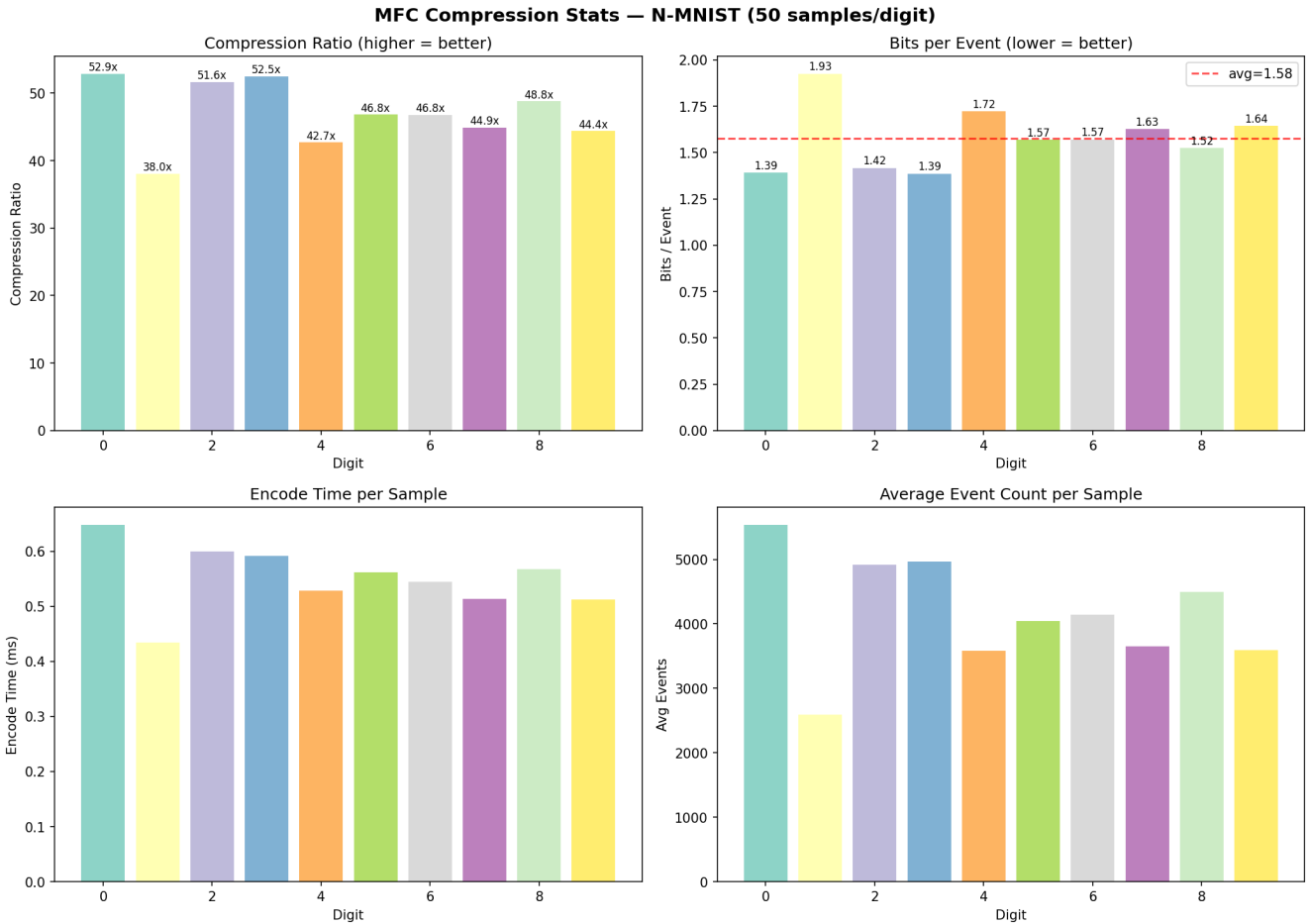


Figure 4: Per-digit compression results aggregated across the N-MNIST training set: compression ratio, bits-per-event, and event counts. The 36–53× envelope across digit classes shows that MFC’s ratio is broadly invariant to digit identity — it adapts to event count and spatial extent without per-class tuning.

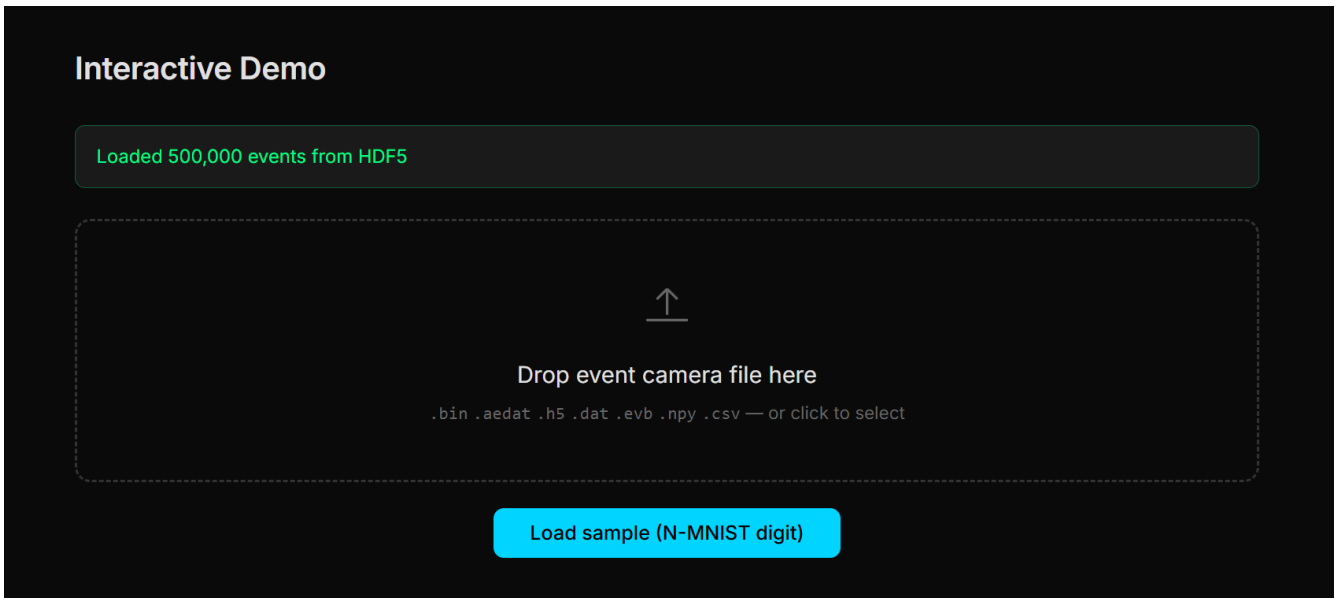


Figure 5: Demo drop-zone at <https://codec.mirrorfractal.com> after loading a 500 000-event window of a DSEC HDF5 recording. The “Loaded 500,000 events from HDF5” banner confirms the WASM build parsed the Blosc/LZ4 HDF5 stream in the browser.

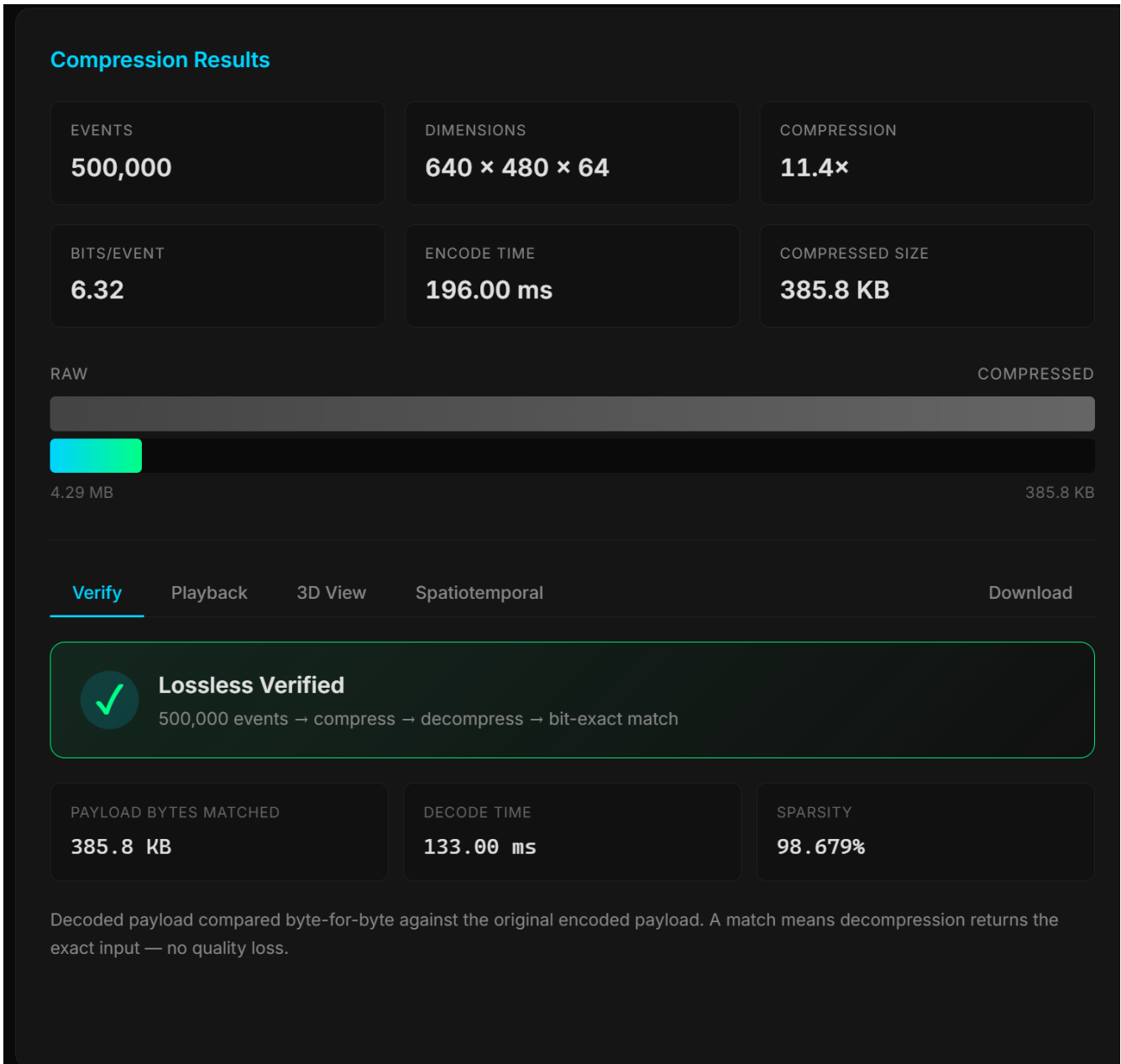


Figure 6: Compression results panel on the same DSEC window: 11.4× ratio, 6.32 bits/event, 196 ms encode, 133 ms decode, 98.68% sparsity, and a green **Lossless Verified** banner asserting a byte-for-byte match between the input and the re-decoded stream. (Numbers differ from Table 2, which reports the specific `zurich_city_01_a` recording; the demo varies with the uploaded file.)

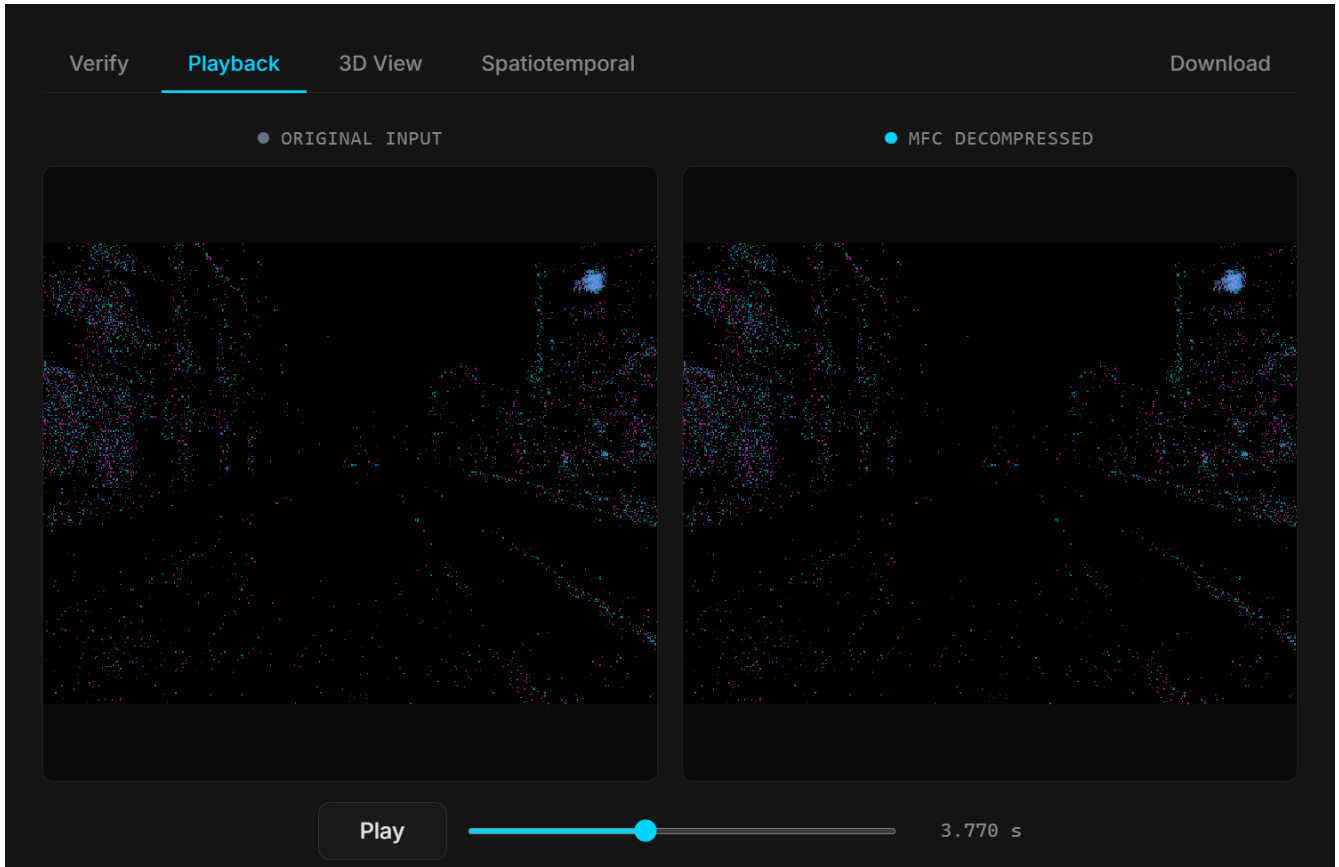


Figure 7: Dual-canvas playback at $t = 3.77$ s: original event stream (left, labelled ORIGINAL INPUT) versus MFC-decoded output (right, labelled MFC DECOMPRESSED). The pixel-identical rendering is the visual counterpart of the Verify tab's byte-for-byte round-trip check.