Lossless Audio Compression:
Progress Report

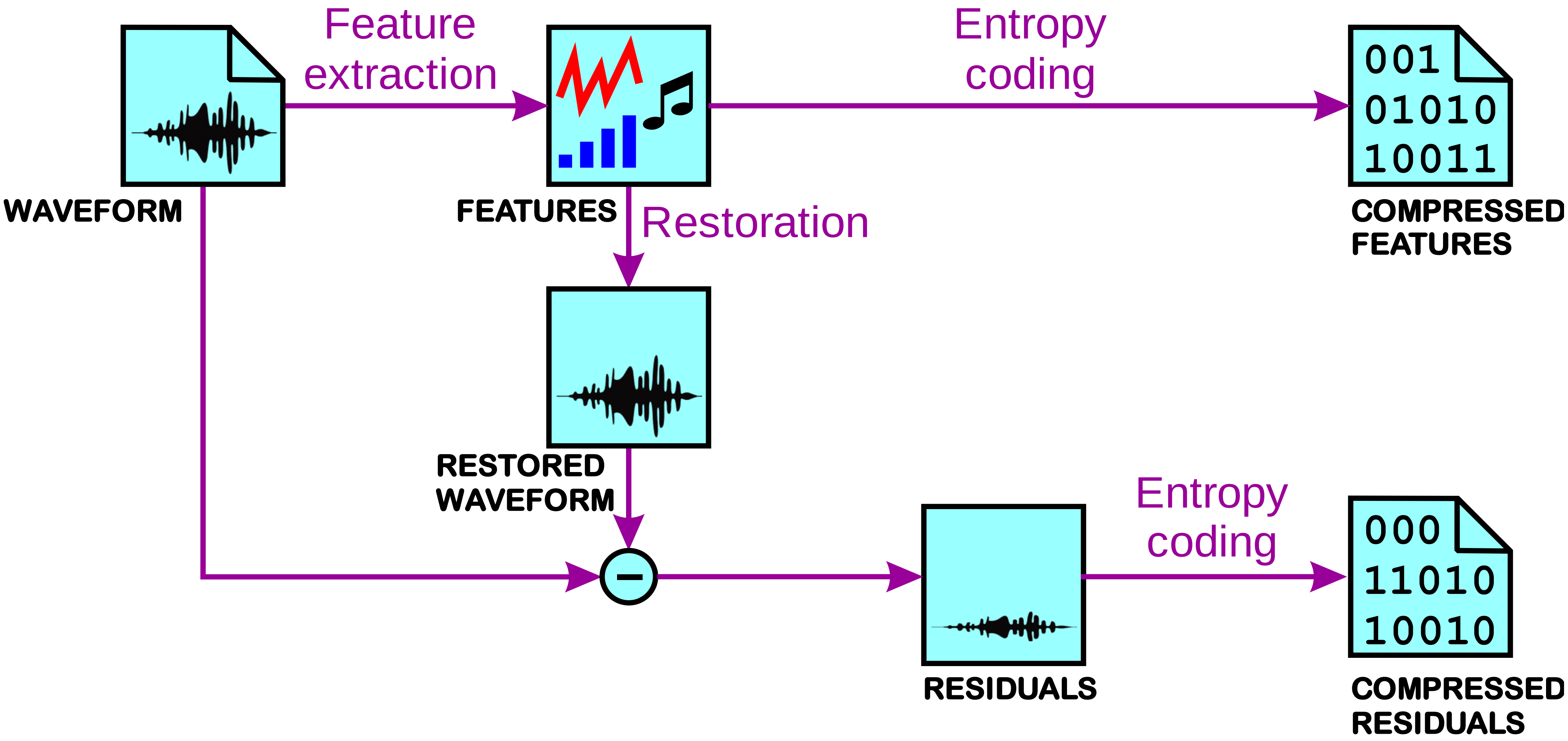
Maribor, 8. 12. 2023

David Podgorelec
Luka Lovenjak
Luka Železnik
Borut Žalik

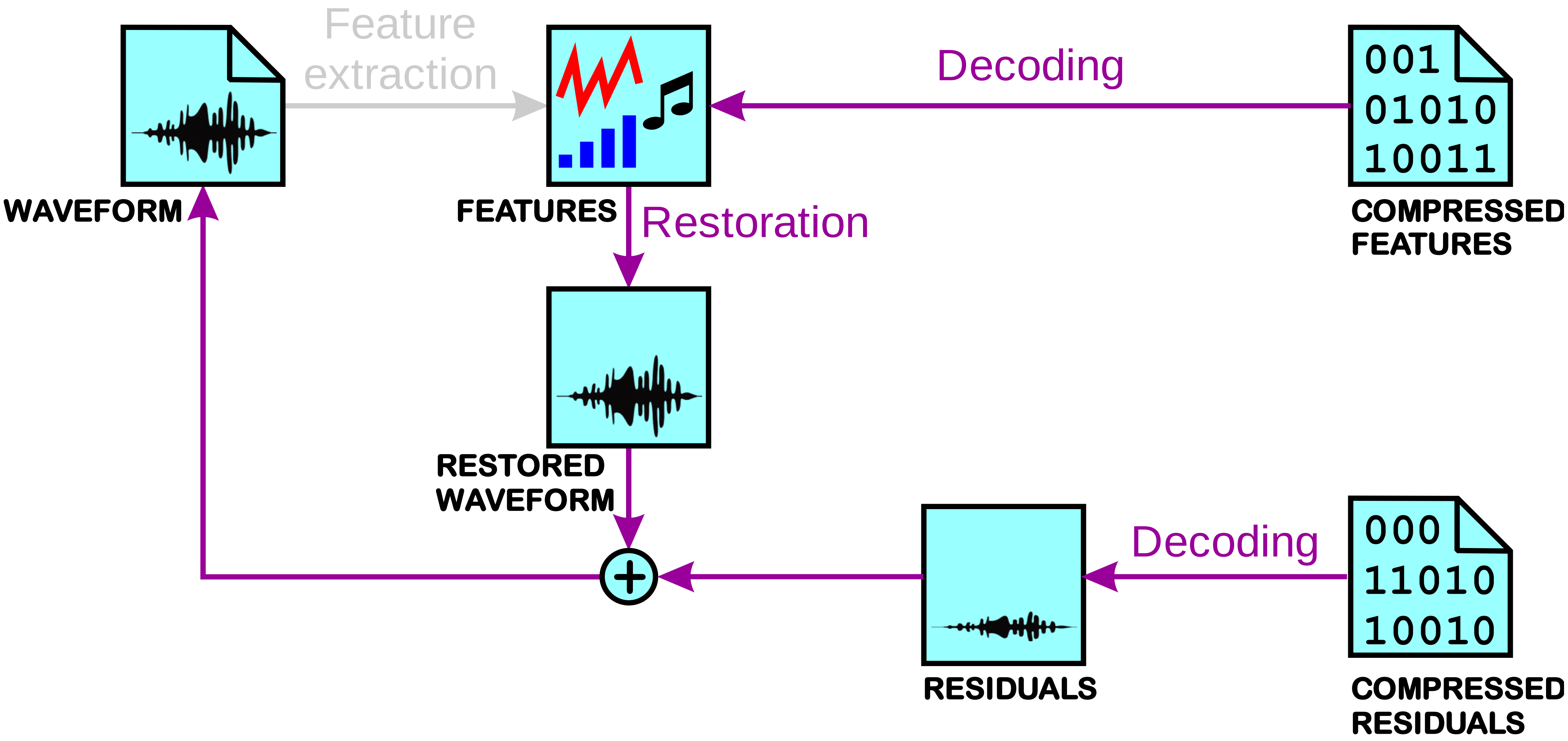
University of Maribor
Faculty of Electrical Engineering and Computer Science

Institute of Computer Science
Laboratory for Geospatial Modelling, Multimedia and Artificial Intelligence

Compromise Lossless Audio Compression



Compromise Audio Decompression



Compromise Lossless Audio Compression

- Feature extraction
  - Feature detection + selection (iterative or heuristic optimisation of CR) + lossy compression (optionally, decompression in the restoration step).
- Restoration
  - Feature-based approximation of the input stream.
- Residuals
  - Differences between the input and the appoximated (restored) waveform.
- Entropy coding
  - Lossless compression, possibly different for features and residuals.
  - $CR = \frac{|INPUT\ WAVEFORM|}{|COMPRESSED\ FEATURES| + |COMPRESSED\ RESIDUALS|}$

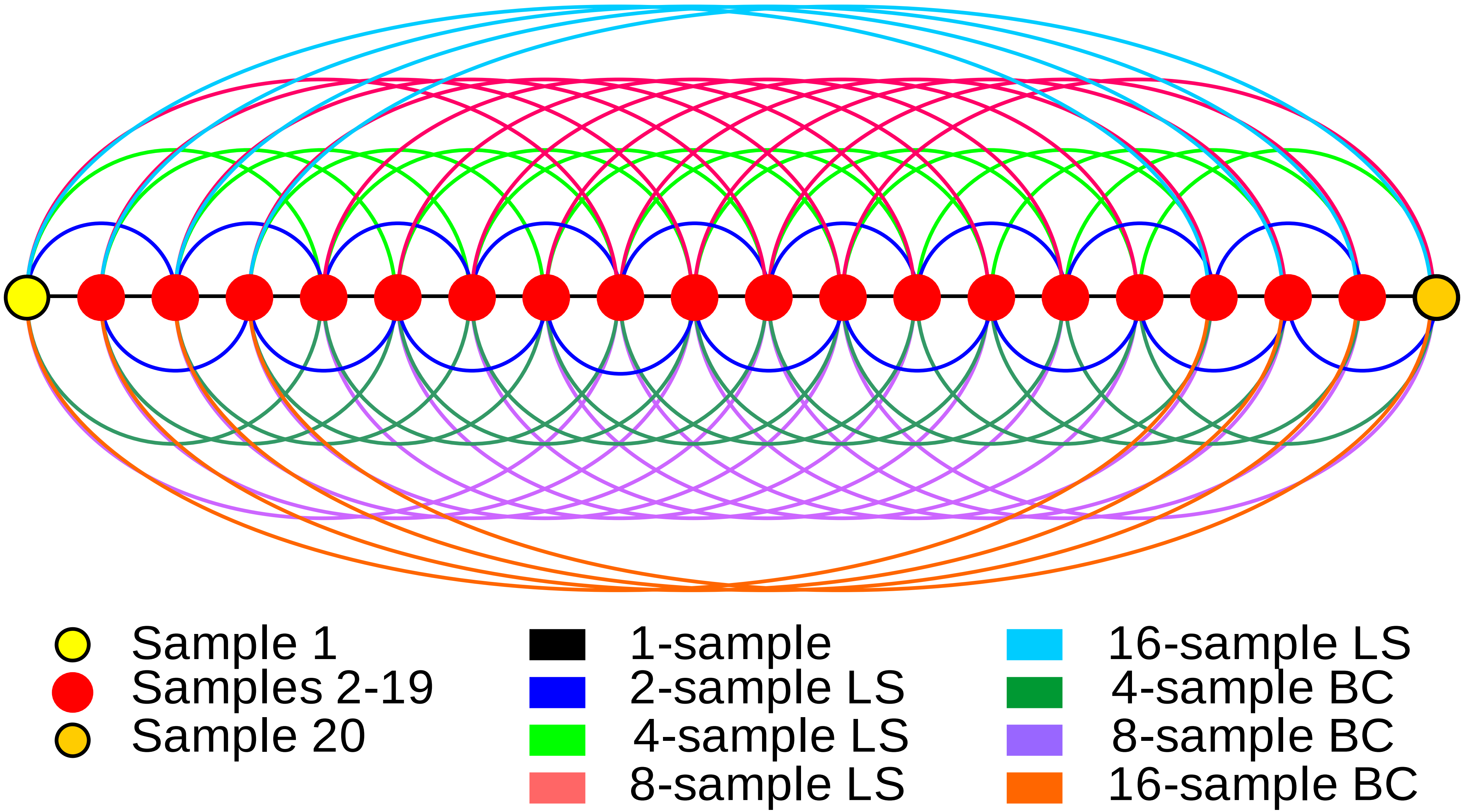
Compromise Lossless Audio Compression

- Four versions (V1 – V4) until now.
- Our input:
  - 44.1 kHz PCM waveform (CD Audio, WAV without a header)
  - 16-bit audio samples in 2's complement.
  - Mono or stereo. Single (left) channel in visualizations.
- State-of-the-Art:
  - FLAC (2001, Josh Coalson, 2003, Xiph.Org Foundation, 2023),
  - MPEG-4 ALS (2006-2009, ISO),
  - Monkey's Audio (APE, 2000-2023, Matthew T. Ashland).

V1: Dynamic Programming

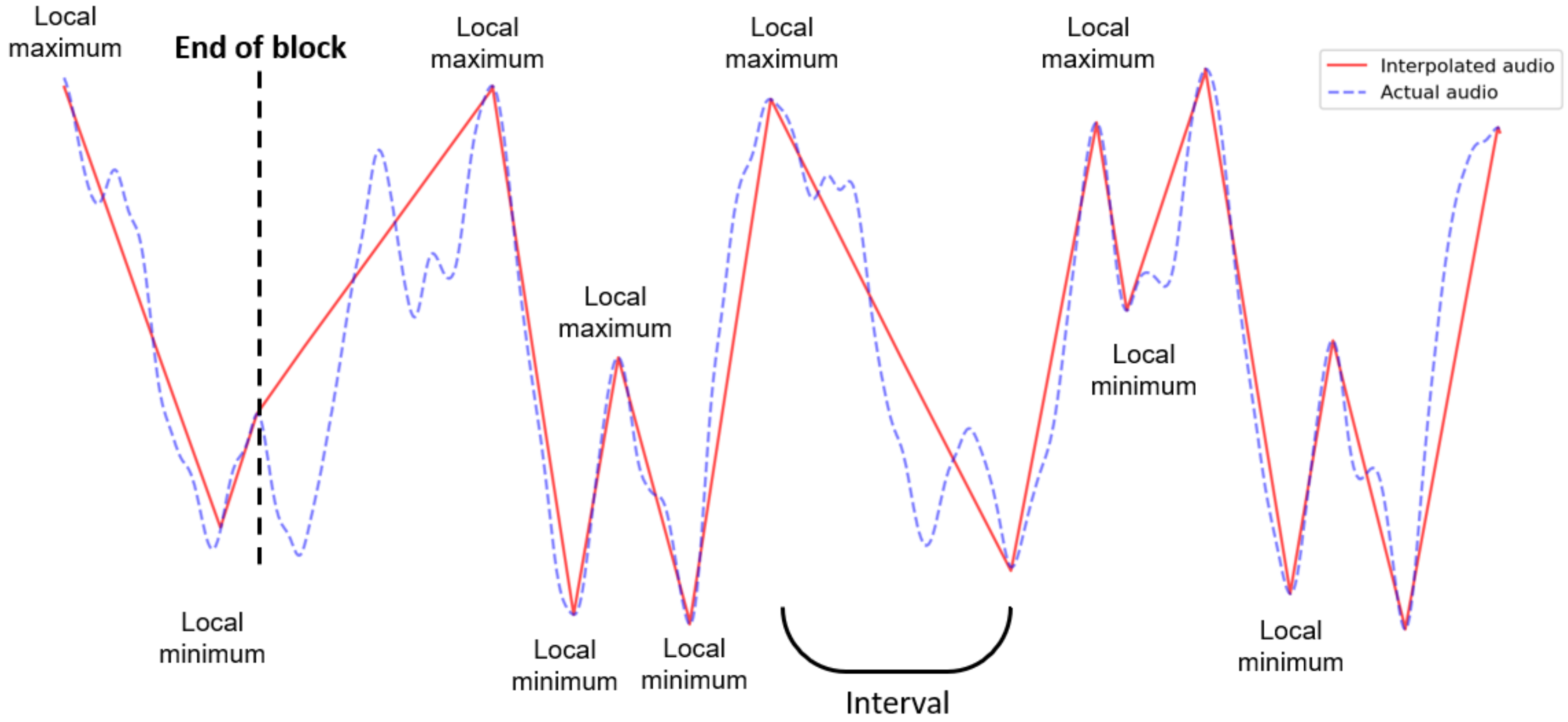
- Feature extraction
  - Predictions of individual audio samples or longer strings of consecutive samples (same in Version 2). String lengths e.g. from {1, 2, 4, …, 512}.
  - Prediction = interpolation (approximation) with a line segment or a quadratic Bézier curve (feature).
  - Feature selection: greedy method or dynamic programming.
  - Splitting into blocks (obligatory with the dynamic programming).
- Entropy coding
  - BASC, Rice, or Golomb-Rice
  - CR lower (worse) for 0.10–0.18 than in FLAC.
- Future work
  - Perhaps near losless and/or lossy compression.

V1


Sample 1
Samples 2-19
Sample 20
1-sample
2-sample LS
4-sample LS
8-sample LS
16-sample LS
4-sample BC
8-sample BC
16-sample BC

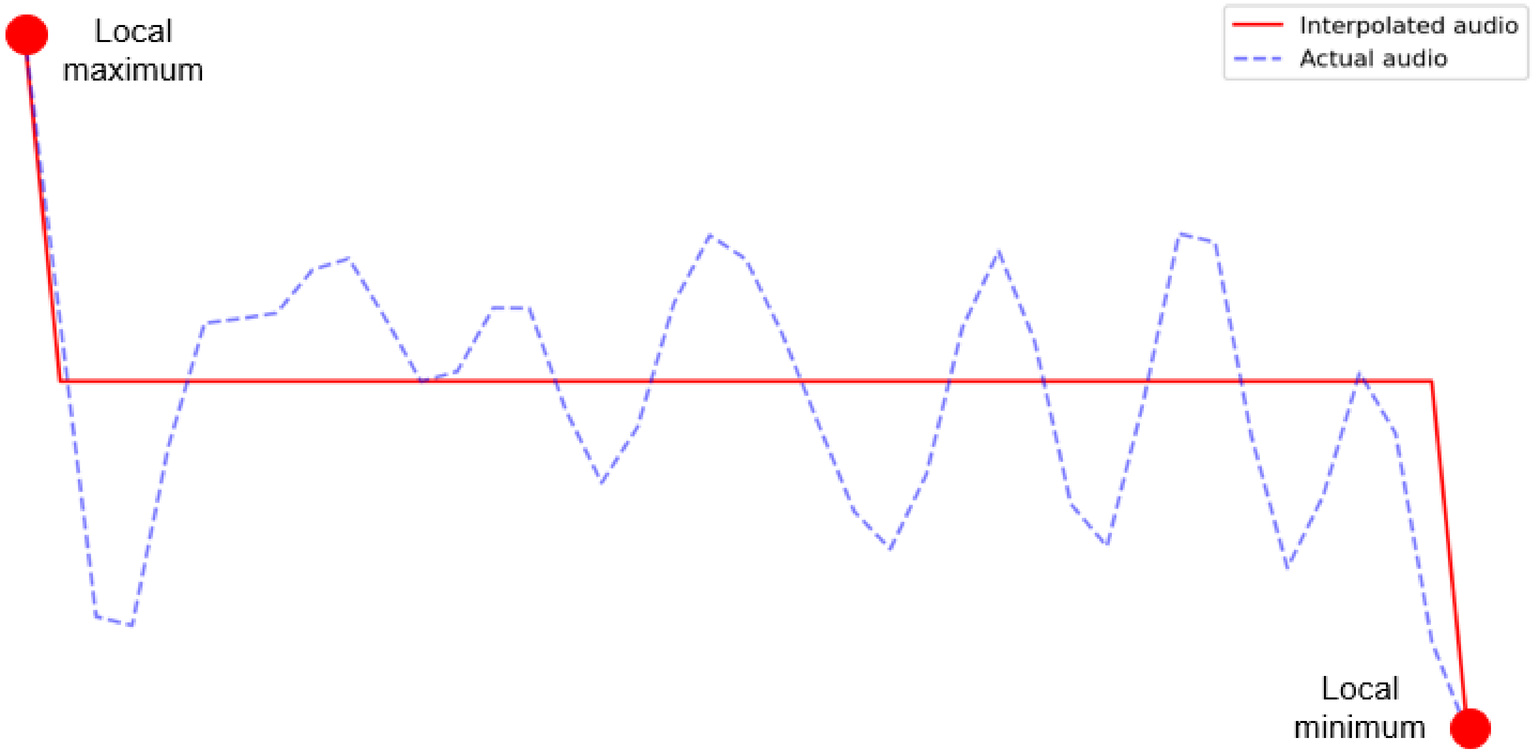
V2: Extreme-Based Interval Approximations

- A feature approximates an interval between two successive distinct extremes.
- 5 approximations:
  - linear,
  - average,
  - grid-based polyline,
  - lossless (verbatim),
  - lossless (RLE).



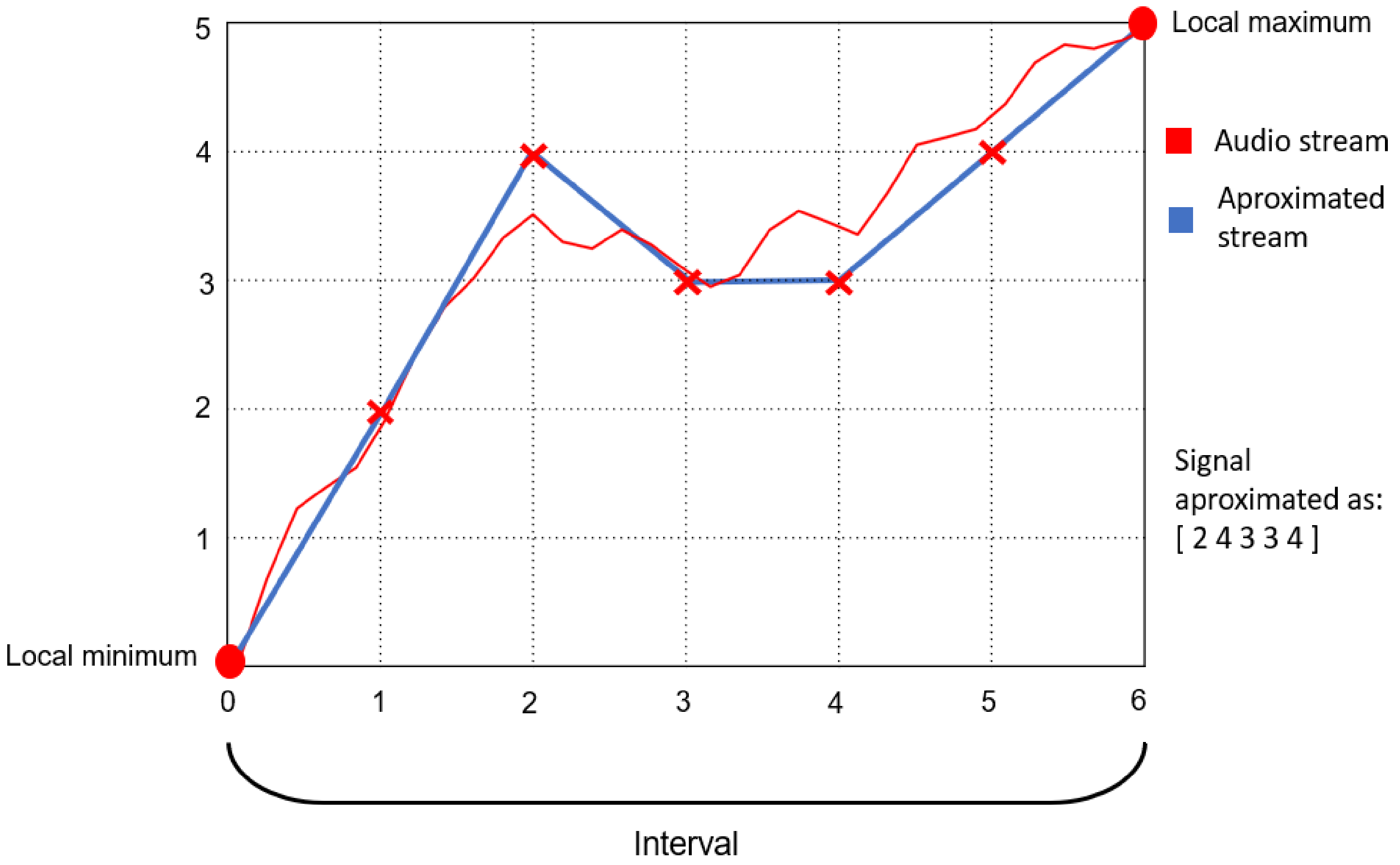
V2: Extreme-Based Interval Approximations

- A feature approximates an interval between two successive distinct extremes.
- 5 approximations:
  - linear,
  - average,
  - grid-based polyline,
  - lossless (verbatim),
  - lossless (RLE).



V2: Extreme-Based Interval Approximations

- A feature approximates an interval between two successive distinct extremes.
- 5 approximations:
  - linear,
  - average,
  - grid-based polyline,
  - lossless (verbatim),
  - lossless (RLE).



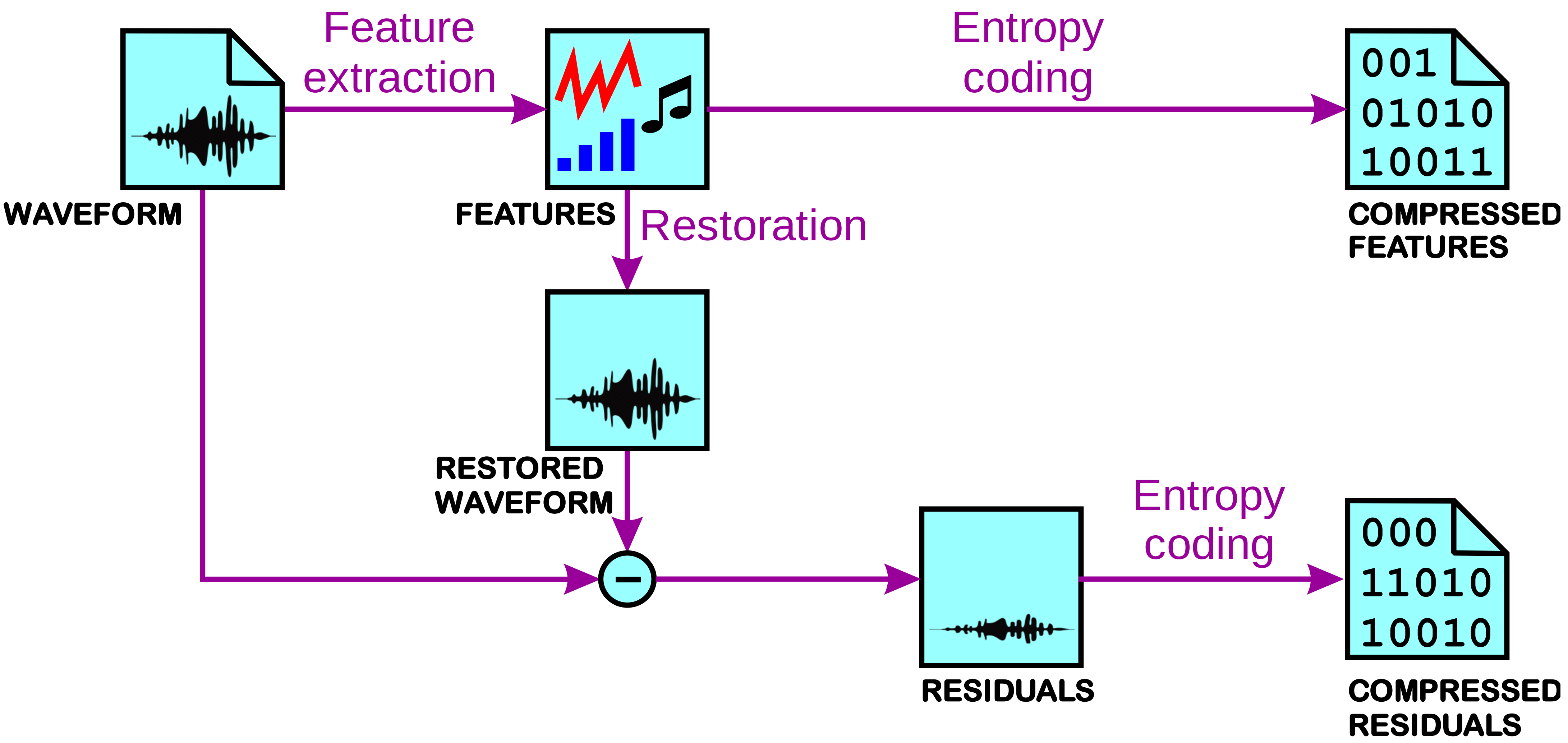
V2: Extreme-Based Interval Approximations

- Entropy coding
  - Features are not too space-consuming compared to residuals. Currently, just a carefully designed compact representation of each feature attribute.
  - BASC, Rice, Golomb-Rice, interpolative coding for residuals.
  - CR lower for 0.05-0.12 than in FLAC (0.10-0.18 in Version 1).
  - CR lower for 0.08-0.16 than in Monkey's Audio (APE).
  - APE is always superior to FLAC, so we use only APE from here on.

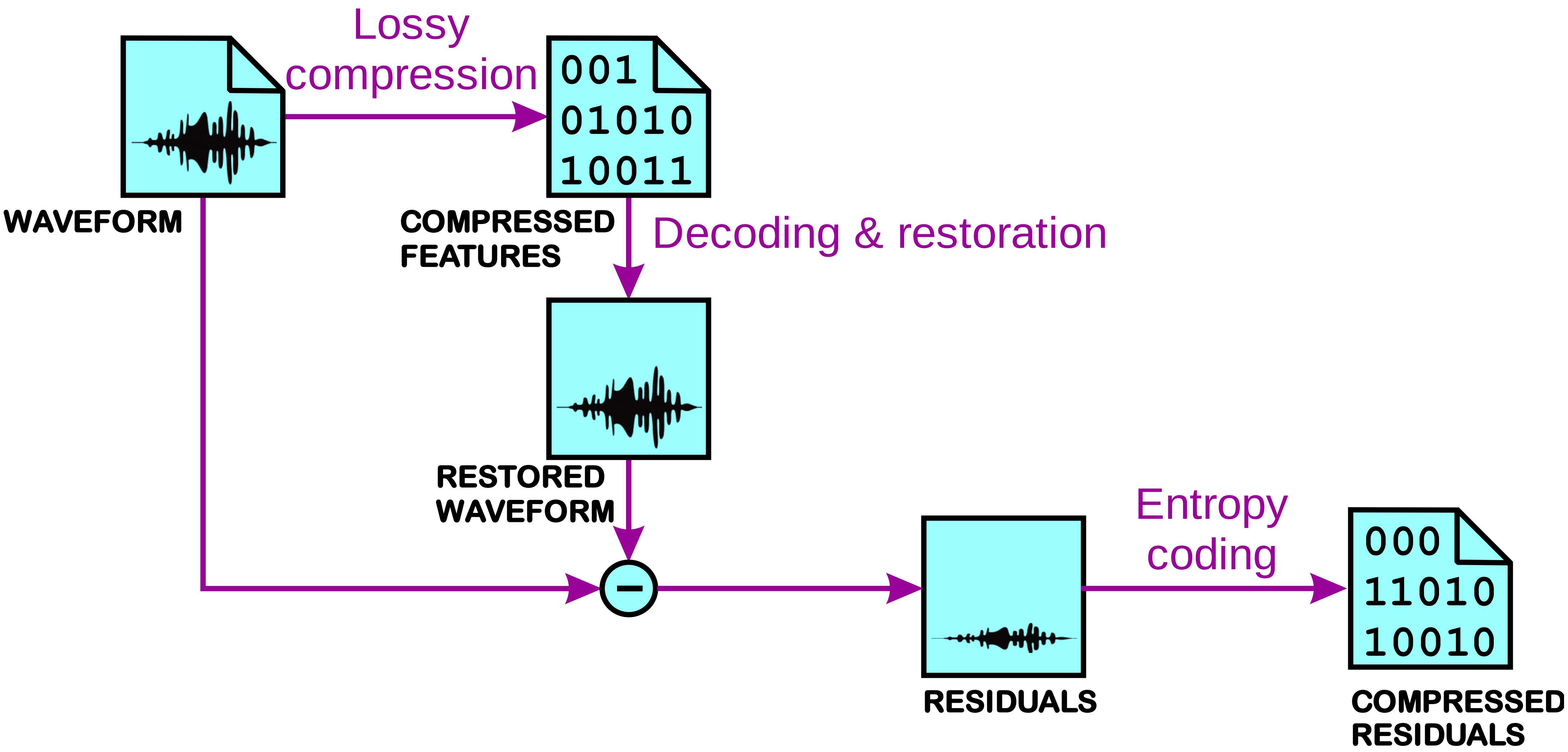
V3: Black Box Features. Is APE even beateble?

- CR may be improved by preprocessing the entropy coding with entropy reduction (MTF, BWT, MwI…), but not enough to beat APE.
- Feature extraction is crucial.
  - More and richer features → smaller and better compressible residuals!
  - But enriching the feature set should not spend too much extra space!
- Feature extraction may be considered lossy compression, as its results enable the restoration of a lossy waveform.
- Versions V3 and V4 are „simple“ experiments where SOTA lossy compression (**black box**) is used for feature extraction.

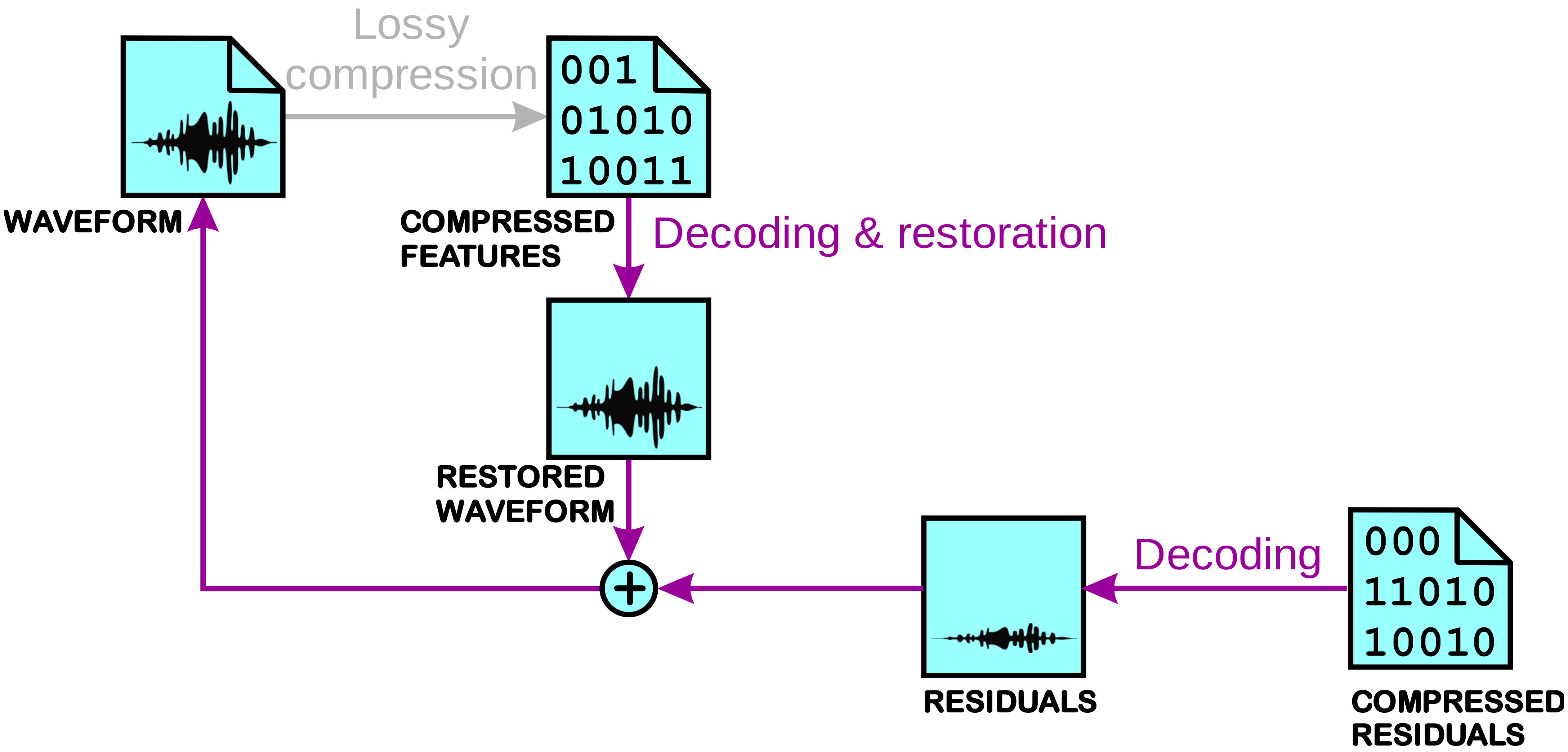
V3: Compression



V3: Compression



V3: Decompression



V3: Test Data

- The Rolling Stones, Anybody Seen My Baby, 00:04:07, stereo.

File	Parameters	[B]	[MB]	CR
WAV	44.1 kHz, 16 bits	43,693,418	41.8	1.00
FLAC	Quality 8 (highest)	30,346,577	28.9	1.44
APE	Quality Insane	29,580,674	28.2	1.48

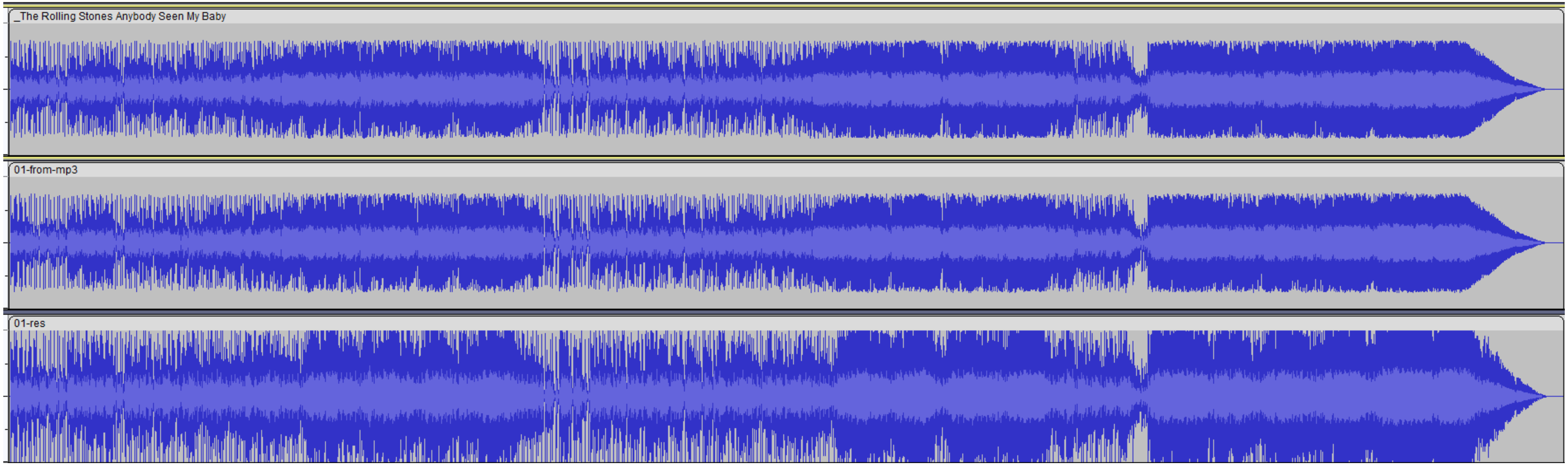
- For automatic tests cca. 20 other songs and shorter excerpts.
- In V3, we use APE for residuals. Can lossy compressed file (features) + APE beat APE alone?

V3: mp3

- The most „natural“choice for the black box feature extraction.
- Why black box? We do not have to deal with entropy coding of individual features neither with restoration (mp3 decompression does it).

- Test 1:
  - Input 41.8 MB (stereo PCM, 1411 kbps)
  - Output (black box of) features: mp3, 320 kbps
  - Output residuals: APE, insane quality

V3: Test 1

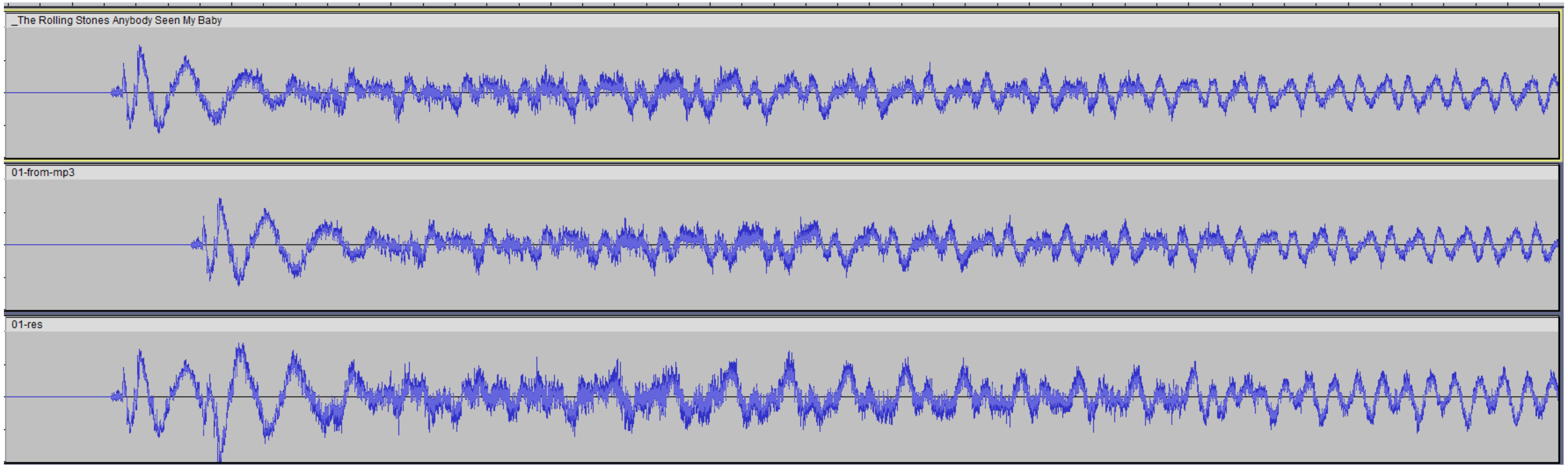


- The input stream at the top, the restored waveform in the middle, and the residuals at the bottom.

V3: Test 1

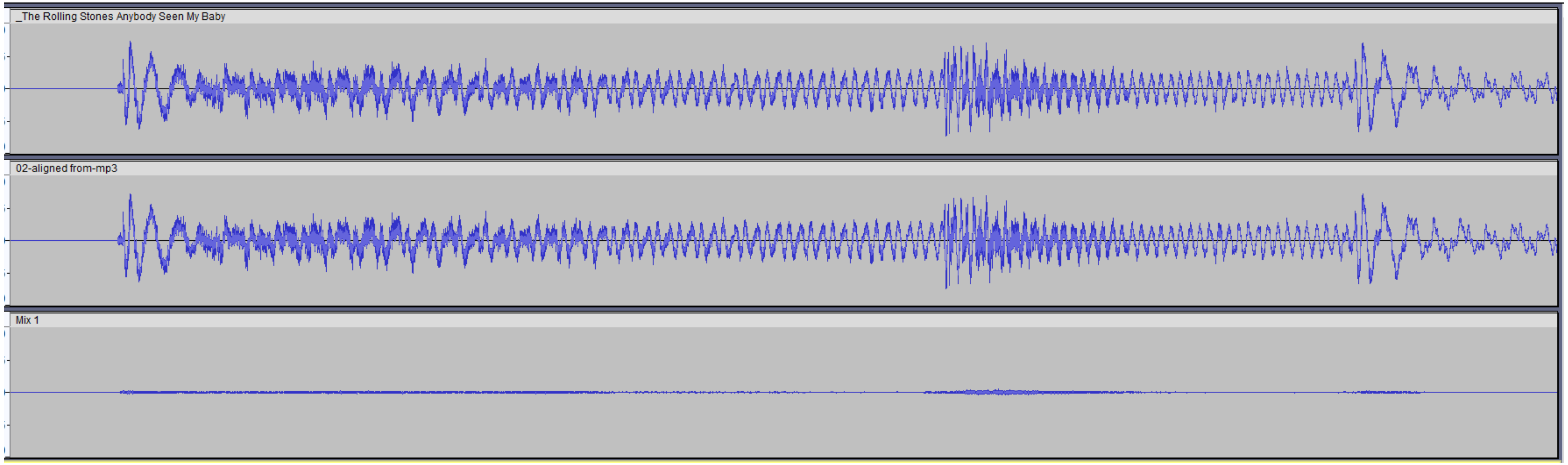
Test	Orig. * [MB]	Orig. CR	Feat. [MB]	Res. [MB]	Feat. + Res. [MB]	Feat. + Res. CR	Comparison CR
V2	28.2	1.48	1.15	30.3†	31.43	1.33	0.15
V3.1	28.2	1.48	8.93‡	29.4*	38.43	1.08	0.40

* APE, † BASC, ‡ mp3



V3: Test 2

- Delay at the beginning of mp3 file is due to silence (0.025055 s) inserted by the algorithm intentionally (why?).
- There is also some silence added (or removed) at the end of the file, bur irelevant for us. Test 2 uses mp3 aligned with WAV.



V3: Test 2

Test	Orig. * [MB]	Orig. CR	Feat. [MB]	Res. [MB]	Feat. + Res. [MB]	Feat. + Res. CR	Comparison CR
V2	28.2	1.48	1.15	30.3	31.43	1.33	0.15
V3.1	28.2	1.48	8.93	29.4	38.43	1.08	0.40
V3.2	28.2	1.48	8.93	22.6	33.08	1.32	0.16

- CR comparable with V2 (extreme-based interval approximation).
- 320 kbps mp3 carries the majority of the audible signal. APE 22.6 MB is thus too big. APE compresses PCM the best but is it suitable for residuals? V4.
- Let first try with mp3 below 320 kbps (smaller black box).

V3: Tests 3-7

Test	kbps	Orig. [MB]	Orig. CR	mp3 [MB]	Res. [MB]	mp3 + res. [MB]	mp3 + res. CR	Comparison on CR
V3.2	320	28.2	1.48	8.93	22.6	33.08	1.32	0.16
V3.3	256	28.2	1.48	7.40	23.74	31.14	1.34	0.14
V3.4	128	28.2	1.48	3.67	26.83	30.50	1.37	0.11
V3.5	96	28.2	1.48	2.69	27.3	30.05	1.39	0.09
V3.6	32	28.2	1.48	0.98	28.07	29.06	1.43	0.05
V3.7	8	28.2	1.48	0.48	28.15	28.63	1.46	0.02

- mp3 8 kbps better than V2, but still worse than APE alone.
- APE of residuals (28.15 MB) close to APE of original (28.2)…

V3: mp3

- Serious drawback is requirement for alignment with WAV.
- Time-consuming. Needed in both, encoder and decoder.
- Delay 0.025055 s for all bit rates, but only in Audacity.
- 0.010905 in VLC Media Player…
- Delay detection is a further slowdown.
- Some other choice for the black box of features?

V3: Vorbis OGG

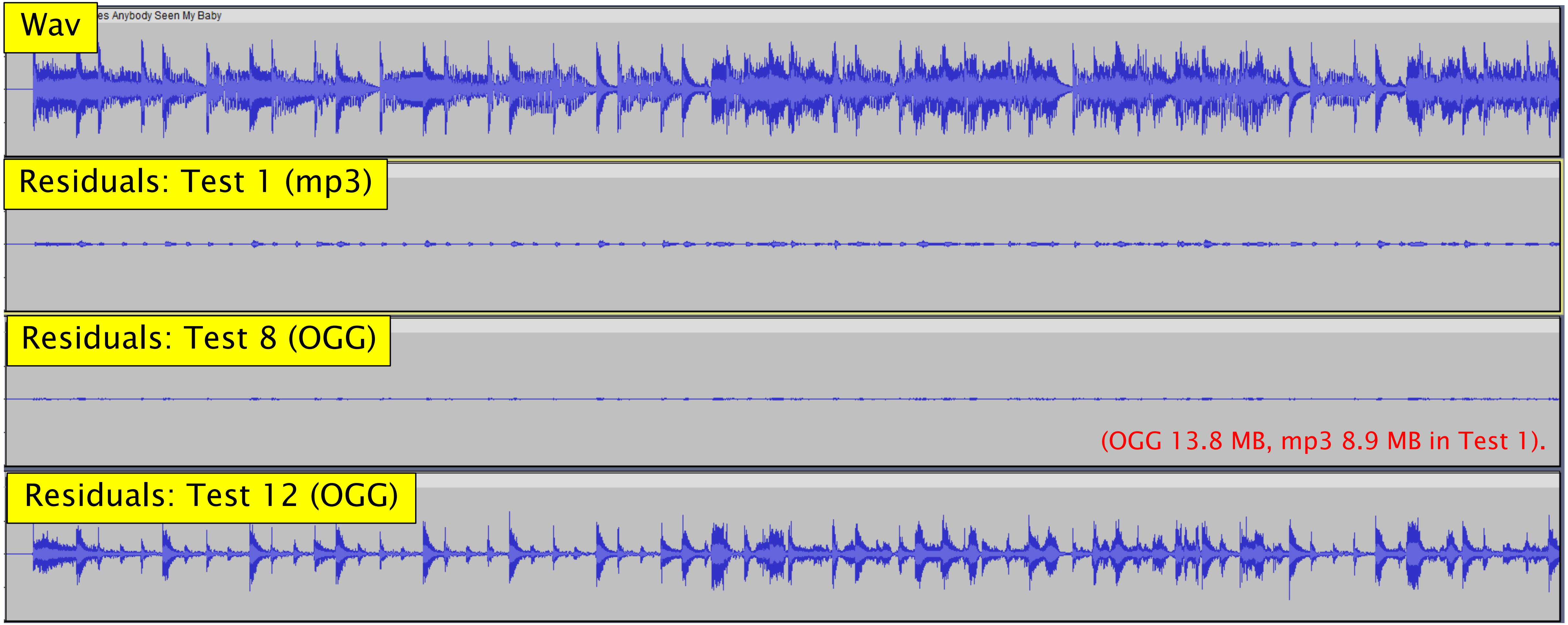
- Xiph.Org Foundation (the same as FLAC).
- 2000, fully open and competitive with mp3.
- Slightly better quality at the same CR.
- Syncronization with the original WAV (no need for the alignment).
- CR manipulated through downsampling (8 kHz – 48 kHz) and the quality parameter (0–10) instead with the bit rate.
- Also based on MDCT (no need for details here, as we use it as a black box).

V3: Tests 8–12

Test	[kHz]	Quality	Orig. [MB]	Orig. CR	OGG [MB]	Res. [MB]	OGG + res. [MB]	OGG + res. CR	Compa-rison CR
V3.8	44.1	10	28.2	1.48	13.80	17.41	31.21	1.34	0.14
V3.9	44.1	0	28.2	1.48	1.85	27.55	29.40	1.42	0.06
V3.10	11.025	6	28.2	1.48	1.83	27.33	29.16	1.43	0.05
V3.11	2	10	28.2	1.48	0.66	28.19	28.85	1.45	0.03
V3.12	2	0	28.2	1.48	0.19	28.26	28.45	1.47	0.01

- With some tracks, slightly better CR achieved than by APE alone!
- compression_results.xlsx

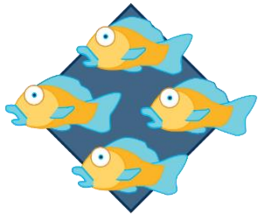
V3: Vorbis OGG



V4: Omitting APE for residual coding

- Own implementations of known entropy reduction and entropy coding algorithms. First attempt with AC and MwI+AC.
- Big expectations, but catastrophic early results ☹ for both, AC and MwI+AC.
  - MwI is an entropy reduction transformation introduced by Žalik. It is similar to MTF, but it transfers the considered sample with some surrounding are to the front instead of the sample alone.
- In best case (downsampling and quality parameters of OGG + additional parameter delta for MwI), CR is 0.25 lower than in APE.
- MwI+AC always achieves better (up to 20%) or at least nearly the same results as AC alone.

V4: What's next?



- So we are back at the beginning again.
- First careful analysis and tests of MwI and AC implementations.
- Try OGG+BASC… or OGG+MwI+BASC…
- Omitting the black box.
  - Try V2+MwI+AC…
  - Some improvements of V2 also waiting for testing. Every bit counts (use some entropy coding for features besides a compact representation).
- Near-lossless and lossy compression.
  - All versions V1-V4 to be considered.