Deterministic Ensemble Decoding for FT8 and FT4: Re-Reading the Same Audio, and the Runtime Split That Pays for It

Tihomir Sokcevic (CE3TSK)
Santiago de Chile
ce3tsk.com github.com/ce3tsk/jtdx_contest

August 2026

Abstract

The FT8 and FT4 decoders in WSJT-X and in its derivatives JTDX and MSHV resolve a received band through a chain of hard decisions taken on soft data: a sync threshold, a strongest-first candidate ordering with successive interference cancellation, several rejection gates, and a belief-propagation decode that stops at the first codeword passing CRC. Each of these flips when the input is perturbed by less than the noise, so repeated runs of a non-deterministic decoder sample different subsets of the messages actually present. We show that this sampling can be made *deliberate and reproducible*. After removing the data races—nine inherited in FT8, four in FT4, two introduced by our own pipeline—so that the decoder returns identical output at any thread count, we re-decode the same audio through a small fixed set of perturbations—a sample delay, seeded additive dither, and a sub-bin frequency shift—and keep the union. Sampling costs runtime, and the scarce resource is not the processor but the reply deadline: the decode cannot begin before 14.1 s into the period and the sequencer must answer about a second later. We therefore split the runtime budget *at* that deadline instead of fitting the work inside it—a short reply-critical phase, then a background phase that runs on through the next period and, after the operator's own transmission, through two. Affordable depth rises by an order of magnitude with the deadline unmoved; on the reference capture the split alone carries the decoder from 78 to 102 of 104 known messages. On a 240-period on-air capture this ensemble, combined with an opposite-recipe pass over the residual and a hint memory extended from one to four same-parity periods, yields 6 208 unique FT8 messages against 4 869 for stock JTDX v2.2.159 with its known-call lookup switched off (27.5 %; 4 802 and 29.3 % as shipped) and 5 249 for the same decoder at JTDX's own recipe (18.3 %), with the reply decision still taken about one second into the decode. On FT4, threaded on the same slice grid, the same perturbation family gives 6.0 % over a baseline that already includes the hint memory, and the full stack—hint memory, ensemble, alternate pass and a TX background with its own settings—reaches 1 888 messages against 1 525 for the stock FT4 decoder with its lookup off (24 %; 39 % as shipped) with the reply decision at 0.59 s of a 1.36 s deadline. Across more than 100 perturbed FT8 runs, every one of the 16 additional messages fell inside an independently validated reference set, with no phantom decodes observed.

1 Introduction

FT8 and FT4 [1] are the dominant weak-signal digital modes on the amateur HF bands. Both transmit a 174-bit LDPC codeword—77 message bits plus a 14-bit CRC—as tones on a fixed grid, and both are designed to decode at signal- to-noise ratios well below what a human operator could copy. In a contest the value of a decoder is not only how deep it reaches but *when* it reaches there: an automatic sequencer must choose its reply before the transmitter keys, which in FT8 leaves roughly one second after the decode begins.

This paper describes the decoder of a JTDX fork prepared for the 2026 World Wide Digi DX Contest, and reports what each of its changes is worth on recorded audio. Our central claim is methodological rather than algorithmic. The decode chain is a sequence of threshold decisions on noisy evidence; its output is therefore a *sample* rather than a deterministic function of the band. Historically this manifested as an embarrassment: non-deterministic builds occasionally reported far more messages than deterministic ones, and the difference was attributed to luck. We argue the opposite reading is correct—the extra messages were real, and the sampling was the mechanism—and that once the decoder is made reproducible, the same sampling can be performed on purpose, at a chosen cost, with the union of the samples as the output.

Contributions. (i) We identify determinism as a *precondition* rather than a hygiene property, and remove the races that prevented it (§3.1). (ii) We split the decoder's runtime budget *at* the reply deadline rather than inside it, so the depth a decoder can afford is bounded by the period and not by the deadline; of the implementations we examined, none schedules work past the deadline in this way (§2.3, §3.5). (iii) We define a three-kind perturbation ensemble and measure the yield of each kind (§3.2, §5). (iv) We show that a pass with the opposite demodulation recipe over the residual recovers messages no perturbation reaches (§3.3). (v) We extend the a-priori hint memory from one to four same-parity periods and quantify both its gain and its phantom-decode cost (§3.4).

2 Background

2.1 Signal structure

An FT8 transmission is 79 symbols of 8-FSK at 6.25 baud, 12.64 s long; 21 symbols carry three Costas sync arrays and

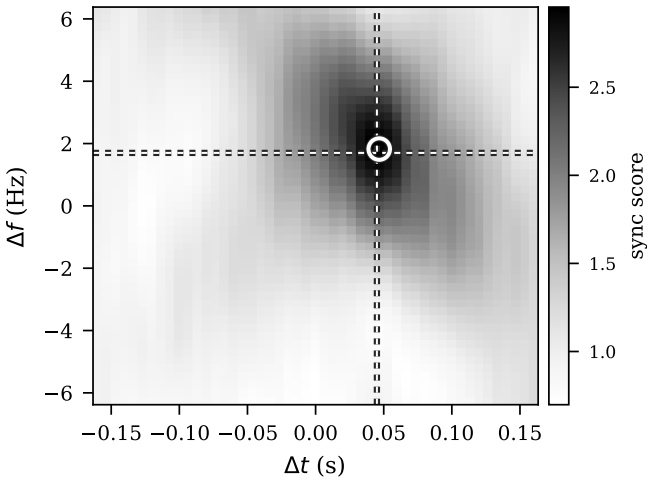


Figure 1: Costas sync score over the alignment plane, simulated FT8 signal at −15 dB SNR in 2 500 Hz, on a 49×49 grid. Dashed lines mark the true offset; the ring marks the search maximum, recovered to within one grid cell. *Simulated*, not measured on air.

the remaining 58 carry the codeword. Tones are spaced at exactly the baud rate, so the phase advance across a symbol is a whole number of cycles and consecutive symbols are phase-coherent. FT4 uses 4-FSK at 23.4375 baud over 105 symbols.

2.2 The stock decode chain

Demodulation produces 174 log-likelihood ratios which are passed to a belief-propagation LDPC decoder, with an ordered-statistics decoder [3] as fallback; the CRC adjudicates. Because verification is cheap and reliable, the decoder can afford to guess repeatedly—a property the rest of this work exploits.

Before demodulating, the stock decoder already searches for a better alignment than the candidate search provided, walking the arrival time over $\pm\frac{1}{4}$ symbol and the frequency over a few hertz, rescoring the Costas sync at each point. Figure 1 shows this surface for a simulated signal at −15 dB: the ridge is narrow, and a candidate misaligned by half a symbol sits on the floor rather than the peak.

The stock decoder also computes bit metrics at three coherent block lengths (one, two and three symbols for FT8; one, two and four for FT4) and tries each in turn. Figure 2 shows why all three are retained rather than only the longest. With a stable phase reference, coherent combining over longer blocks wins as theory predicts. Under random-walk phase noise the ordering *inverts*, and the single-symbol metric becomes the best of the three. Since the decoder cannot know in advance which regime a given signal is in, it computes all of them.

2.3 Scheduling in existing implementations

Every implementation we examined runs one decode per received period and leaves the decoder idle between them. WSJT-X and JTDX trigger the decode when the period's audio is complete and the decoder process then waits for the next trigger. MSHV divides its FT8 decode into three passes, but all three fall inside the receive period; the last is triggered at 14.4 s and nothing runs after it [4]. In each case the whole decode must finish before the sequencer's reply decision, so the depth that can be afforded is bounded by the deadline rather than by the period. Instrumenting our own build, the decoder process was idle for roughly 12 of every 15 seconds—in transmit and receive periods alike.

This is the constraint §3.5 removes, and it is what makes the ensemble of §3.2 affordable: a three-member ensemble costs 7.6 s, which no schedule bounded by a one-second deadline can spend.

3 Method

3.1 Determinism as a precondition

Nine data races were found and removed in the stock multi-threaded FT8 decoder, and four more in FT4 once it was given the same slice threading; two further races introduced by our own pipeline were found and removed in the same campaign. The most consequential was a shared audio buffer from which threads subtracted decoded signals, so that what any thread observed depended on scheduling order; the others involved a downsampling cache, an OSD initialisation flag, an FFT plan-cache publication, waveform tables, the decoded-message store, and—found later, while making the threaded output byte-identical—the arrival-ordered emission and hint feed, mid-pass hint retirement, and shared a-priori tone tables that upstream had flagged as suspected memory corruption. The band-slice grid was additionally decoupled from the thread count. In both modes the decoder now returns identical output—messages, SNRs, time offsets and provenance markers—at any thread count ≥ 2.

This is what makes the ensemble an instrument rather than a lottery: perturbing the input is only informative if leaving it unperturbed is reproducible. Independently, two hot loops were rewritten—a sync window sum hoisted out of a lag loop, and the OSD inner loop repacked into 64-bit words—reducing a single-threaded reference decode from 14.6 s to 6.8 s with output proven identical on more than 39 000 randomised cases.

3.2 The perturbation ensemble

An ensemble member re-decodes the *pristine* band—not the residual—after one reproducible perturbation, and prints only messages not already found. Three kinds are used:

- **Delay.** The band is shifted by n samples with zero fill. Reported DT is corrected by $-n/12000$.
- **Dither.** Gaussian noise at $k\times$ the band RMS, drawn from a private `xorshift64*` generator with a fixed seed, deliberately not the Fortran intrinsic whose sequence is compiler-defined. The useful range is 3 % to 5 % of RMS; at 30 % the weak tier is destroyed.

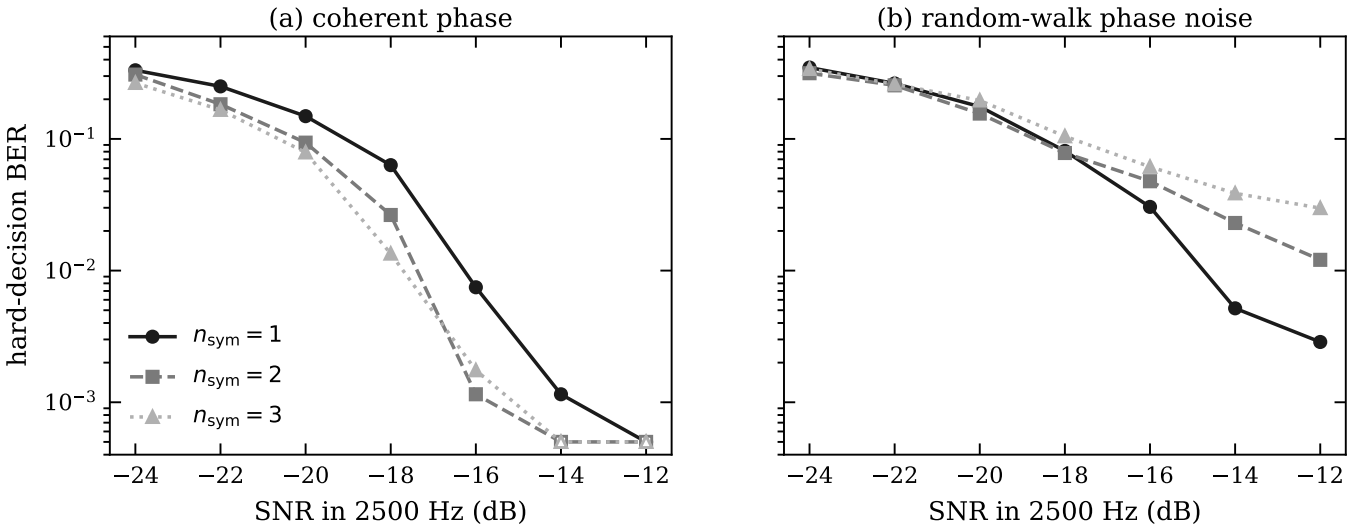


Figure 2: Hard-decision bit error rate against SNR for the three coherent block lengths, from a Monte Carlo reimplementation of the metric arithmetic. Ten transmissions per point (≈1 740 bits); hollow markers denote no errors observed, not zero error rate. Bits are hard-decided directly from the metric to isolate it, whereas the deployed decoder passes them to belief propagation—the vertical scale is therefore relative. *Simulated*, not measured on air.

- **Frequency shift.** The analytic signal is formed by FFT, multiplied by $e^{j2\pi\Delta f t}$, and the real part taken. Reported frequencies are corrected by $-\Delta f$.

Members run as additional passes inside a single period's decode. They deliberately do not contribute to the stored signals used for cross-period averaging: feeding perturbed copies of a period in as its own predecessor produced a message that was never transmitted, because averaging near-identical copies is coherent gain that does not exist on the air.

3.3 The alternate pass

JTDX carries two demodulation recipe families—averaged data with a wide DT window, and raw data with deeper ordered-statistics search—which fail on different signals. The alternate pass runs the opposite family once over the residual after the normal passes have subtracted what they can.

3.4 Extended hint memory

The stock *hint* decoder re-tries a previously decoded message at a candidate's frequency and DT with all 77 payload bits fixed, using the existing a-priori machinery, but only if that message was decoded in the immediately preceding same-parity period. A single missed period therefore breaks a station's chain. We extend the memory to four same-parity periods, searching older lists only when newer ones hold nothing and retiring an entry once used.

3.5 Splitting the runtime budget

Two fixed times set the shape of this. The decode cannot start before 14.1 s into the period whose audio it decodes, because that is when the audio it needs has arrived; and the auto-sequencer needs its reply decision about one second later, as the next period—the one the operator transmits in—begins. Anything slower than that is too late to change what is sent, however good it is.

The usual reading of those two numbers is that the decoder has one second. That is true only of the work whose result the sequencer needs. The rest of the period is unclaimed, and so is the operator's own transmission. We therefore cut the decode at the deadline rather than inside it:

- an **RX phase**, deliberately cheap—five cycles, sensitivity 2, one ensemble member—which finishes about 1.1 s in and emits the decodes the sequencer needs, exactly as an unsplit decoder would;
- a **background phase** on the retained band, running the units the RX phase could not afford: the deferred alternate pass, the ensemble members, a residual pass with every known message subtracted and the threshold lowered, and a plain non-SWL decode of the pristine band.

The budget is what changes. Bounded by the deadline, a decoder may spend about a second; bounded by the period, it may spend about thirteen. When the next period is one the operator transmits in, the arithmetic improves again: no new audio arrives to be decoded, so that period's decode is skipped and the background phase simply continues—two periods, close to 29 s, rather than one. The deadline has not moved and the reply is unaffected either way.

Correctness at the boundary is by explicit request rather than by timing. Each unit re-reads the request counter and a lock file before it starts and aborts *within* a pass when the next decode is due, so a long unit cannot delay the next period's RX phase; only messages not already printed are emitted, carrying the period's own UTC, and a closing line reports the count. Decodes made in the background are marked in the output so an operator can tell them from reply-time decodes.

Table 1: Unique FT8 messages over the on-air hour, same audio, each engine at its own best settings, 12 threads where the engine has them. Stock rows are the real v2.2.159 binary with the file-decoding driver restored; the baseline is stock with its ALLCALL7.TXT known-call lookup switched off, its honest best - "as shipped" keeps the lookup, which alone rejects 67 real decodes.

Engine	Messages	vs. stock
This work, pipeline ensemble full	6208	+27.5%
This work, pipeline ensemble	6194	+27.2%
This work, recommended preset	6091	+25.1%
This work at JTDX's own recipe	5249	+7.8%
WSJT-X 3.0.2 (multi-thread)	5171	+6.2%
WSJT-X improved 3.2.0	5163	+6.0%
Stock JTDX v2.2.159, no ALLCALL7	4869	—
Stock JTDX v2.2.159 as shipped	4802	−1.4%
WSJT-X 3.0.2 (standard, depth 3)	4782	−1.8%

The split also compounds with §3.4. Messages the background recovers enter the same-parity hint lists, so a station it finds late is available as 77-bit a-priori knowledge to the RX phases that follow: the next deadline-bound second is spent on stations still unknown rather than on re-deriving one already heard. We report the two mechanisms separately in Table 3 and do not attempt to attribute the interaction between them.

4 Experimental setup

Two corpora are used. The *benchmark* is a single wideband capture with an independently validated 104-message reference set, used for the FT8 member selection of Fig. 3. The *on-air hour* is 240 consecutive 15 s periods recorded 2026-08-27, 16:57–17:57 UTC on 20 m over 100 Hz to 3 100 Hz, used for all engine comparisons; FT4 results use a separate 240-period hour of 7.5 s periods recorded 2026-08-29 from 23:38 UTC on 14.080 MHz over the same range, with a 73-period day hour as a check. All measurements ran on 12 threads of an otherwise idle AMD Ryzen 7 5800H; the two captures differ, so the modes are not comparable to each other. Every engine was given its own best settings, and each configuration was replayed from the same recorded audio.

Both hours and the wideband benchmark capture are published as 16-bit WAV, with their reference sets and a cross-platform script that replays them through the standalone decoder and reports these same quantities, at `https://ce3tsk.com/download/wav/`.

5 Results

Ensemble. Adding five members takes the benchmark from 85 to 101 messages (Fig. 3), at 2.6 s to 12.8 s. The first two members are inexpensive recipes and carry 10 of the 16 gained. The three-member setting (98 messages, 7.6 s) is the shipped contest target.

Alternate pass. On the same capture the alternate pass alone raises the best non-ensemble recipe from 81 to 85 messages and the weakest-tier count from 7 to 8 of 17, while being *faster* than the configuration it replaces (5.7 s against 7.2 s), because it displaces an older half-slice offset pass whose extra subtractions removed what it would have found.

Pipeline. On the benchmark capture the RX phase alone resolves 78 of the 104 reference messages; the background phase raises that to 102, a gain of 30.8 % and the largest of any single mechanism in Table 3. On FT4 the TX background is worth 8.3 % over a baseline that already carries the hint memory. Neither figure costs anything at reply time: the RX phase is unchanged in both measurements, and the FT4 reply decision stays at 0.59 s of its 1.36 s window.

Hint memory. Figure 4 shows the depth sweep on the on-air hour. Depth 4 gives 5.9 % on the classical recipe and 6.0 % on the light preset at no measurable time cost. Between 86 % and 89 % of the gained messages are chains continuing through a missed period, and the yield per additional period of memory falls monotonically but has not reached zero at eight.

FT4. Figure 5 shows the same perturbation family applied to aligned FT4 candidates. Six members give 5.4 % over a baseline that already contains the four-period hint memory, and 6.0 % with the alternate pass. Of the 111 messages the 2026-08-30 run gained, the two frequency members contributed 59, the delays 30 and the dither 22. Notably this is *not* a sync-grid resolution effect: reducing the coarse frequency step of the candidate search gains nothing. The candidates were already being found; what a member changes is the demodulation of one the decoder had abandoned.

The ensemble is one layer of the FT4 stack. FT4 is threaded on the same anchored 417 Hz slice grid as FT8, each slice subtracting into its own copy of the audio, which took six members from 131 % of the 1.36 s reply deadline to 39 %. It then received the pipeline: a TX background switched on exactly as FT8's, which re-decodes the retained period while the station transmits with the members the reply phase did not run, a deeper ordered-statistics decode, the alternate pass, a residual pass and lower thresholds, under its own clock until 0.5 s before the window ends; six presets in FT8's tiers; a *budget auto* member count for the max effort tier, chosen before each decode from the learned cost of the whole receive phase so that it stays inside a 1.3 s budget; and a virtual candidate at the QSO frequency worth about 1 dB on the one reply a QSO is waiting for. Table 2 gives the tiers on the night hour.

6 Discussion

6.1 Why perturbation works

The gains do not come from a better estimator. They come from the fact that the chain's decisions are near-threshold for exactly the signals one cares about, and that a perturbation

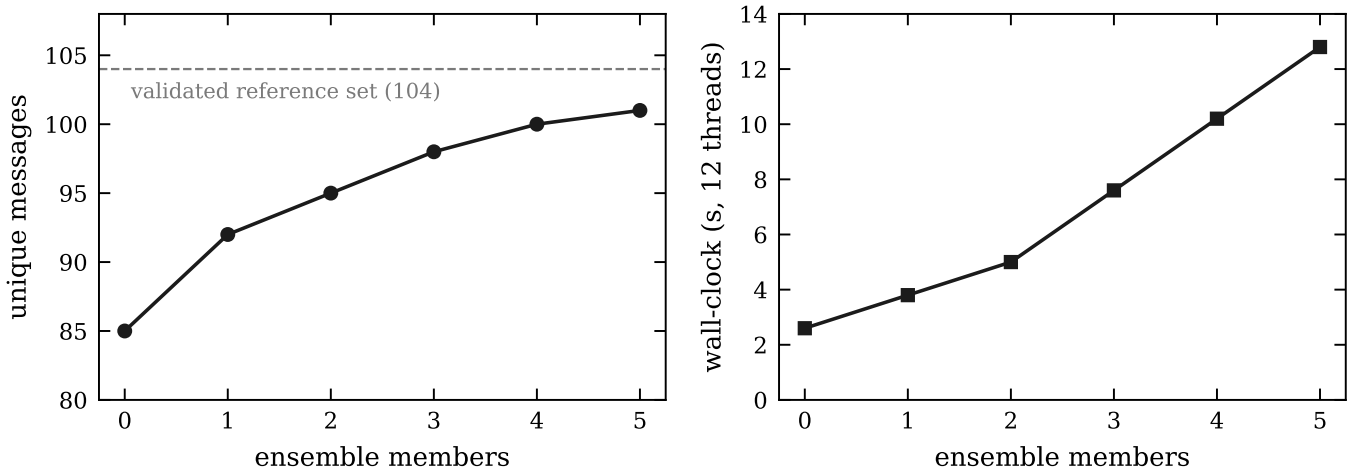


Figure 3: FT8 ensemble on the benchmark capture: unique messages and wall-clock cost against member count, members ordered by measured yield per second. All 16 messages gained lie inside the validated reference set. *Measured*; the member list is fitted to this capture (§6.3).

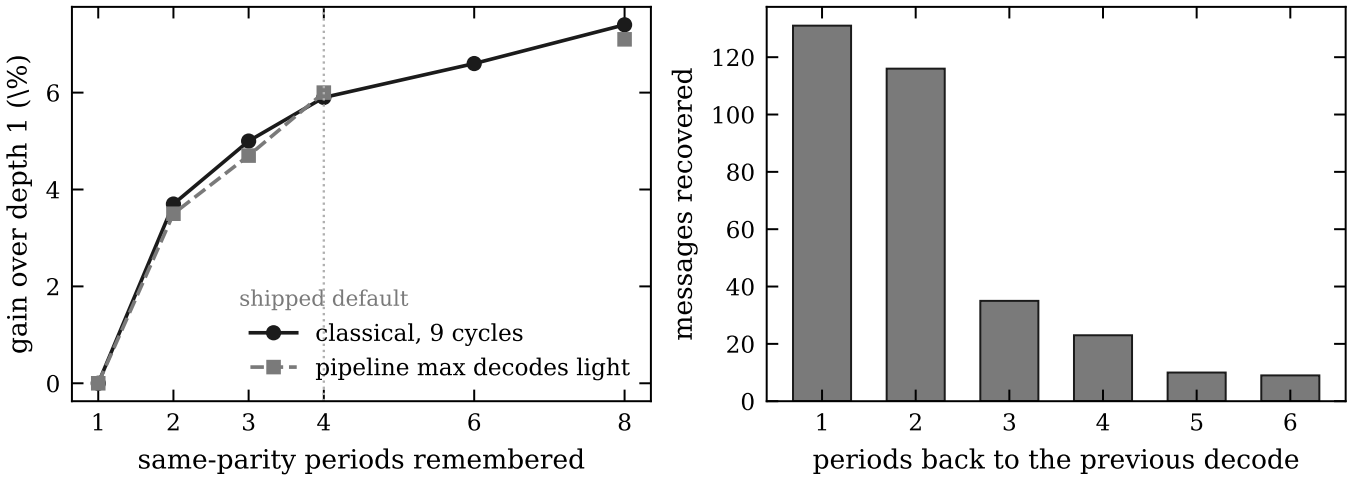


Figure 4: Hint-memory depth. Left: gain over depth 1 for two recipes (the light preset has no depth-6 run). Right: messages recovered by how many same-parity periods had elapsed since the station was last decoded, classical recipe at depth 6. *Measured* on the on-air hour.

smaller than the noise moves different signals across different thresholds. Concretely: three callers within 12 Hz of one another were recovered only by frequency-shifted members, because they sat on a bin edge of the sync grid; two pairs separated by 6 Hz to 20 Hz decoded only under one particular subtraction order. Neither is reachable by tuning a threshold, because no single threshold value is right for both members of such a pair.

6.2 Phantom decodes

Any mechanism that fixes payload bits a priori can manufacture a codeword that passes CRC by chance; with 14 parity bits the per-attempt false-accept rate is roughly $1/16384$, and the ensemble makes many attempts per interval. Two empirical results bound this. First, across more than 100 perturbed FT8 runs, all 16 additional messages fell inside the validated reference set and none outside it. Second, the deeper hint memory produces one to two *stale* hint decodes per hour at depth 4—the previous message on a re-used frequency, sharing enough bits with the new one to pass—rising to three per hour at depth 8. Against 320 to 425 genuine additional decodes this is about 0.05 % of output, and it is the reason the memory stops at four periods rather than eight, where the measured gain was still increasing.

6.3 Threats to validity

The FT8 member list of Fig. 3 is a greedy selection fitted to a single capture; its ordering, and possibly its composition, should be expected to shift on other bands and conditions. The engine comparison uses one hour of one band on one antenna. The two simulated figures are reimplementations of the metric arithmetic rather than instrumented runs of the deployed code, and their vertical scales are relative. FT8 and FT4 measurements come from different captures on different days and bands, so their percentages must not be compared with each other. Finally, the reference set against which "no phantom decodes" is asserted is itself a union of decoder outputs plus cross-checks against a callsign database; it is a

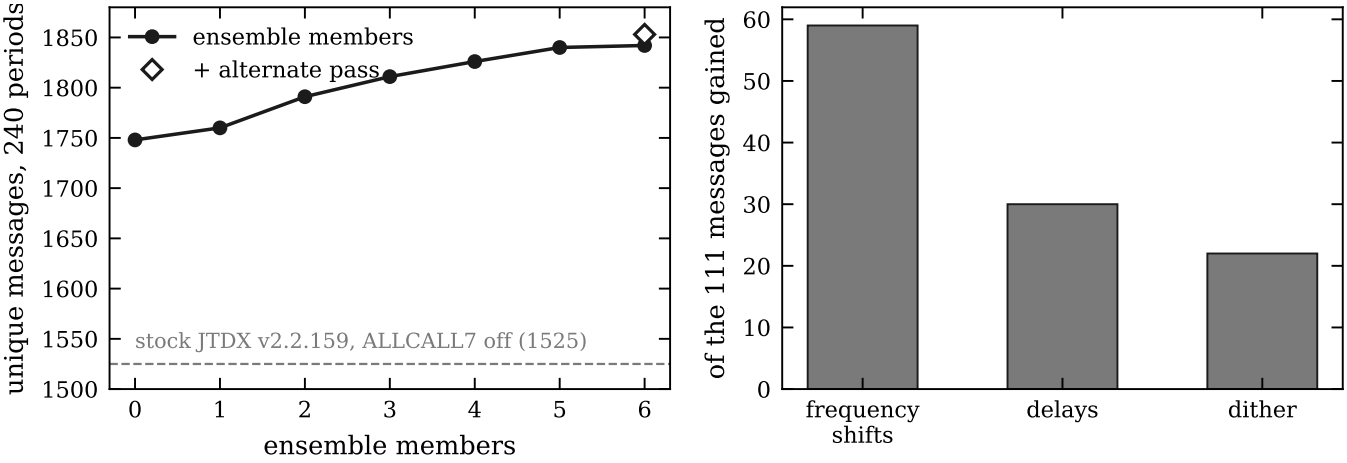


Figure 5: FT4 over a 240-period on-air hour at 12 threads. Left: unique messages against member count, with the alternate pass added. Right: attribution of the gained messages by perturbation kind, from the 2026-08-30 run (111 gained then, 94 on the rerun). *Measured.*

Table 2: FT4 presets over the 240-period night hour, 100 Hz to 3 100 Hz, 12 threads. "Reply" counts the messages decoded before the transmitter keys; the rest arrive in the TX background. Stock is the real JTDX v2.2.159 FT4 decoder (single-threaded; its seconds are wall time per period) with and without its ALLCALL7.TXT lookup, which rejects 169 real decodes on this hour; the WSJT-X rows likewise wall time.

Preset	Msgs	Reply	RX s	Bg s
Stock v2.2.159, shipped	1356	1356	1.35	—
Stock, no ALLCALL7	1525	1525	1.40	—
WSJT-X 3.0.2	1561	1561	0.14	—
WSJT-X impr. 3.2.0	1688	1688	0.22	—
Default	1737	1737	0.10	—
Best power	1822	1761	0.10	0.3
Recommended	1881	1763	0.10	2.1
Most at reply time	1878	1837	0.48	0.5
Max effort	1888	1855	0.59	0.6

Table 3: Contribution of each mechanism, measured separately, both modes at 12 threads; the modes are measured on different captures and are not comparable to each other.

Mechanism	Mode	Measured	Gain
Hint memory	FT8	$5249 \rightarrow 5564$	$+6.0\%$
Hint memory	FT4	$1552 \rightarrow 1737$	$+11.9\%$
Ensemble	FT8	$85 \rightarrow 98$	$+15.3\%$
Ensemble	FT4	$1748 \rightarrow 1842$	$+5.4\%$
Alternate pass	FT8	$81 \rightarrow 85$	$+4.9\%$
Alternate pass	FT4	$1842 \rightarrow 1853$	$+0.6\%$
Full stack, vs. JTDX's recipe	FT8	$5249 \rightarrow 6208$	$+18.3\%$
TX background	FT4	$1737 \rightarrow 1881$	$+8.3\%$
Pipeline	FT8	$78 \rightarrow 102$	$+30.8\%$
Full stack, vs. stock, lookup off	FT8	$4869 \rightarrow 6208$	$+27.5\%$
Full stack, vs. stock, lookup off	FT4	$1525 \rightarrow 1888$	$+23.8\%$
Full stack, vs. stock as shipped	FT8	$4802 \rightarrow 6208$	$+29.3\%$
Full stack, vs. stock as shipped	FT4	$1356 \rightarrow 1888$	$+39.2\%$

strong check but not ground truth.

6.4 Cost

None of this is free, and the deployment reflects that: in FT8 ensemble depth is gated by available threads; in FT4 the presets place the members and the alternate pass in the TX background, where they cost nothing at reply time, and the max effort tier fits its member count to a budget learned from the band; and the pipeline exists in both modes precisely so the additional seconds are drawn from CPU that was previously idle rather than from the reply deadline.

7 Conclusion

Treating a weak-signal decoder as a sampler rather than a function, and then sampling deliberately, recovers a substantial number of real messages that no single run finds—27.5 % on FT8 and, with the TX background, 24 % on FT4 over the stock decoder at its honest best (its known-call lookup off) on recorded on-air audio; 29 % and 39 % over stock as shipped, 18 % on FT8 against the same decoder at JTDX's own recipe, with no observed phantom decodes and with the contest reply deadline preserved. The prerequisite is determinism: the same technique on a racing decoder is indistinguishable from luck, which is how the effect went unrecognised for as long as it did. What makes the depth affordable is where the time is taken from: splitting the budget *at* the reply deadline rather than inside it turns the twelve idle seconds of every fifteen into decoding time and leaves the deadline exactly where it was.

Provenance

All measurements are reproducible from the author's measurement scripts. The corpora themselves—both on-air hours, the wideband benchmark capture and its 104-message reference set—are published at `https://ce3tsk.com/download/wav/` together with a replay script, so the engine comparisons here can be repeated on any machine; the

full measurement harness is not part of the published source repository. The finer grained decode-cycle control referenced in the deployment is inherited from upstream JTDX (2025-09-09) rather than original to this work. This software is a derivative of JTDX by UA3DJY and ES1JA, itself derived from WSJT-X by K1JT and collaborators.

References

[1] S. Franke, B. Somerville and J. Taylor, "The FT4 and FT8 Communication Protocols," *QEX*, July/Aug. 2020.

[2] S. Franke and J. Taylor, "Open Source Soft-Decision Decoder for the JT65 (63,12) Reed–Solomon Code," *QEX*, May/June 2016.

[3] M. P. C. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inform. Theory*, vol. 41, no. 5, pp. 1379–1396, Sept. 1995.

[4] Reference implementations examined for scheduling behaviour: *WSJT-X* 3.0.2 (K1JT and the WSJT Development Group), *JTDX* v2.2.159 (UA3DJY, ES1JA), *MSHV* v2.76 (LZ2HV). Source distributions, 2026.

[5] T. Sokcevic, *JTDX_contest – Improvements for WW Digi 2026*: `IMPROVEMENTS_WW_DIGI_2026.pdf`, the measured inventory of every decoder and contest change. 2026.