



HAL
open science

Low-Latency Software Polar Encoders and Decoders for Short Blocklengths

Mathieu Leonardon, Mohammed El Houcine Ayoubi, Adrien Cassagne, Romain Tajan, Camille Leroux

► To cite this version:

Mathieu Leonardon, Mohammed El Houcine Ayoubi, Adrien Cassagne, Romain Tajan, Camille Leroux. Low-Latency Software Polar Encoders and Decoders for Short Blocklengths. International Symposium on Topics in Coding (ISTC), Aug 2025, Los Angeles, United States. <hal-05120329v1>

HAL Id: hal-05120329

<https://hal.science/hal-05120329v1>

Submitted on 19 Jun 2025 (v1), last revised 4 Jul 2025 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

I - ISTC'25 Contest

- **Challenge Scope:** Develop C++ encoder/decoder implementations for short blocklength codes.
- **Target Platform:** CPU execution, single-threaded.
- **Evaluation Criteria:** Latency (average and maximum) and SNR (E_b/N_0) for frame error rates (FER) of 10^{-3} and 10^{-5} of encoder and decoder implementations.
- **Test Conditions:** Fixed testing setup on an AWGN channel.
- **Submission Requirements:** Implementations must cover at least one full row or column of the following test configurations (code parameters (N, K)):

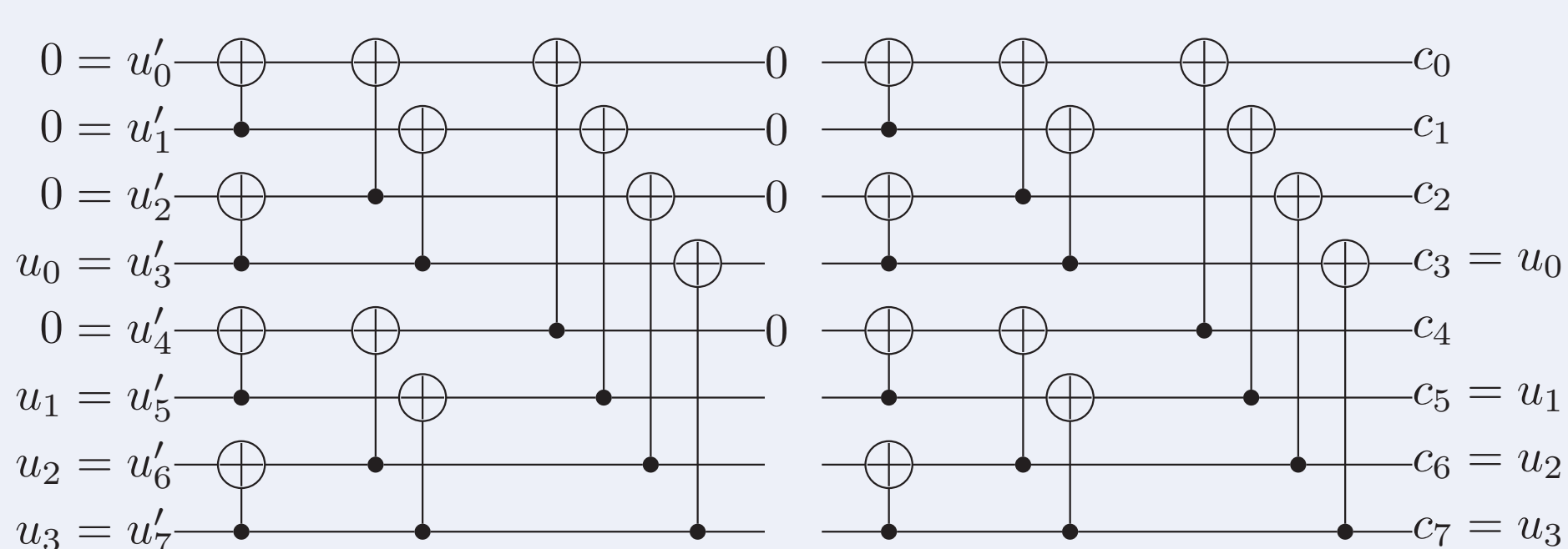
Test configurations.

Rate	K=64	K=128	K=256	K=512
1/4	(256,64)	(512,128)	(1024,256)	(2048,512)
1/2	(128,64)	(256,128)	(512,256)	(1024,512)
4/5	(80,64)	(160,128)	(320,256)	(640,512)

II - Proposed Method

- **Polar Code [1] Decoding:** Adaptive Successive Cancellation List decoding [2] for low-latency and robust error correction.
- **Unrolled Decoder Generator:** A novel tool generates specialized C++ code for unrolled decoders, eliminating control-flow overhead for critical latency.
- **Optimized Bit-Packed Encoder:** Features an optimized encoder implementation with bit-packing and SIMD for efficiency.
- **Extensive Design Exploration:** Various code constructions, CRC selections, and list sizes to find optimal lat./perf. trade-offs.

III - Optimized Encoder



- **Input Vector Generation:** Generate U'_N from frozen and information bits. Statically identify constituent nodes at compile time, skip rate-0 nodes, and vectorize large rate-1 nodes during memory copy.
- **Bit-Packing with SIMD Instructions:** Bit-pack U'_N using `_mm256_cmpgt_epi8` and `_mm256_movemask_epi8` (MSBs extraction).
- **Unpacking with Lookup Table:** Unpacking of the encoded data using a 256-entry lookup table of 256-bit values.

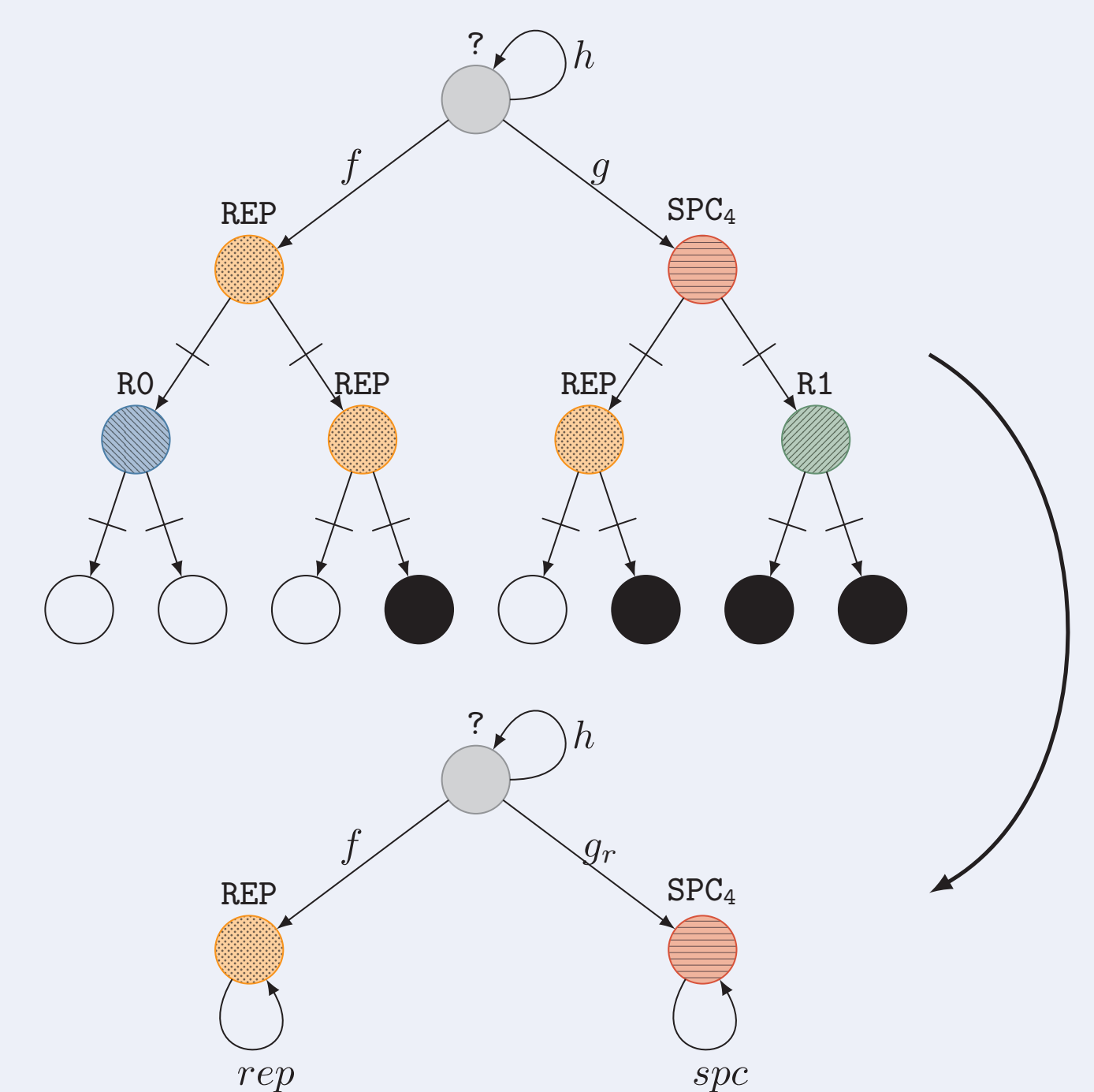
IV - Fully Unrolled Decoders

- **Low Latency Goal:** Aims to achieve very low decoding latency.
- **Source Code Generation:** One Specialized C++ source file for each Polar code configuration.
- **Eliminates Overhead:** Removed loops and recursive calls cost, crucial for short blocklengths.
- **Open Source Tool:** Can generate SC, SCL, and ASCL decoders with tree pruning feature.

```

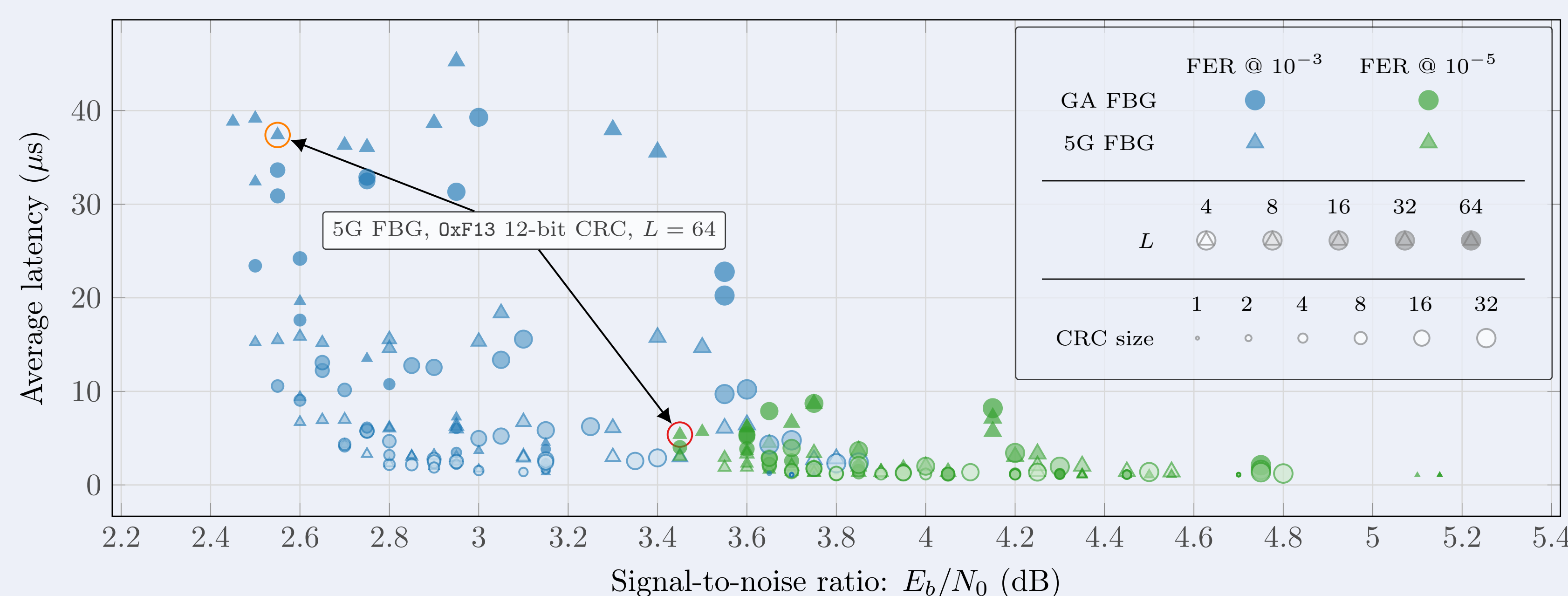
1 // example of a generated simple SC decoder
2 std::vector<bool> frozen_bits_GA_8_4 = {
3 1, 1, 1, 0, 1, 0, 0, 0};
4
5 template <typename API_polar>
6 class Decoder_polar_SC_fast_sys_N8_K4
7 // ... stuff not detailed here
8 void _decode()
9 {
10     API_polar::f <4>( 1, 0+0, 0+4, 0+8);
11     API_polar::rep <4>(s, 1, 8+0, 0+0);
12     API_polar::gr <4>(s, 1, 0+0, 0+4, 0+0, 0+8);
13     API_polar::spc <4>(s, 1, 8+0, 4+0);
14     API_polar::h <4>(s, 0+0, 0+4, 0+0);
15 }
16 };

```



V - Search Space Exploration

Exploration conducted on the Intel Ultra 9 185H processor with 16 cores (1 p-core @ 5.1 GHz used, others left unused). **Uncovering “sweet spots” that balance latency and error-correction.**



VII - Conclusion

- **Open source encoders and decoders** for ISTC'25 challenge [3].
- Specifically targeting **fast implementations** for short blocklengths on CPUs.
- **Toolchain based on Polar codes [4]:** Optimized bit-packed encoder and a **fully unrolled ASCL decoder generator**.
- **Design space exploration:** Configurations balancing latency and decoding perf.
- Open source implems are **available within AFF3CT [5]** to foster reproducibility and further research in channel coding.

VI - Latency & Performance Results

Proposed codes and the corresponding achieved performance.

Rate	K	Code specifications			Encoder $\mathcal{L}_{\text{worst}}$ (μs)	Decoder				
		CRC size	CRC poly.	Fb. gen.		FER @ 10^{-3}		FER @ 10^{-5}		
						L	E_b/N_0	\mathcal{L}_{avg} (μs)	E_b/N_0	\mathcal{L}_{avg} (μs)
1/4	64	11	0x385	5G	0.20	64	2.15 dB	2.53	3.35 dB	0.55
	128	12	0xF13	5G	0.28	64	1.60 dB	6.44	2.60 dB	1.07
	256	12	0xF13	5G	0.60	64	1.25 dB	10.58	2.25 dB	1.66
	512	12	0xF13	GA	1.01	32	1.10 dB	14.92	1.85 dB	3.53
1/2	64	7	0x65	5G	0.14	32	3.15 dB	0.70	4.60 dB	0.33
	128	12	0xF13	5G	0.21	64	2.55 dB	3.31	3.45 dB	0.69
	256	12	0xF13	5G	0.34	64	2.15 dB	6.55	3.10 dB	1.18
	512	16	0x8005	5G	0.68	64	1.90 dB	11.95	2.70 dB	2.12
4/5	64	8	0x9B	GA	0.15	32	5.05 dB	0.48	6.45 dB	0.16
	128	10	0x3D9	5G	0.20	32	4.40 dB	1.21	5.35 dB	0.41
	256	10	0x3D9	5G	0.34	32	4.10 dB	2.20	5.00 dB	0.67
	512	12	0xF13	5G	0.66	32	3.75 dB	4.92	4.55 dB	1.36

Links & References

- [1] E. Arkan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE TIT*, 2009.
- [2] M. Léonardon et al. Fast and flexible software polar list decoders. *Springer JSPS*, 2019.
- [3] M. Léonardon et al. ISTC'25 contest source code. https://github.com/bonben/istc25_contest.
- [4] M. Léonardon et al. Polar decoder generator. https://github.com/aff3ct/polar_decoder_gen/tree/modif_istc25.
- [5] A. Cassagne et al. AFF3CT: A fast forward error correction toolbox! *Elsevier SoftwareX*, 2019.

SCAN ME

