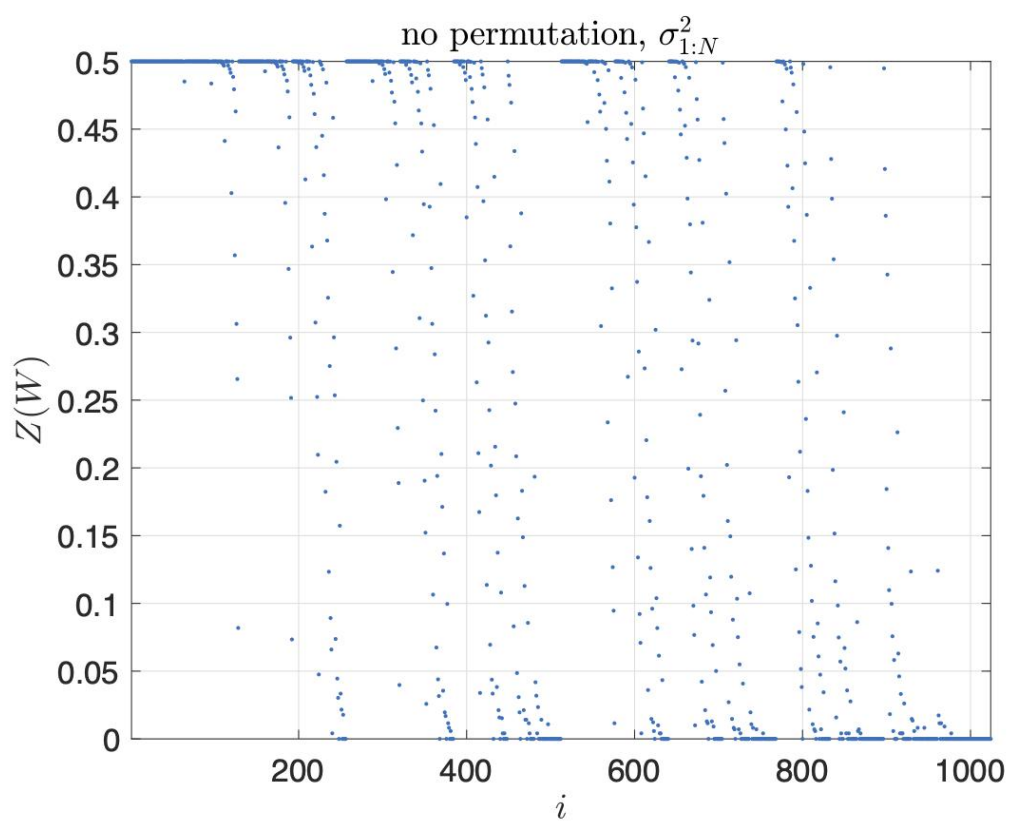
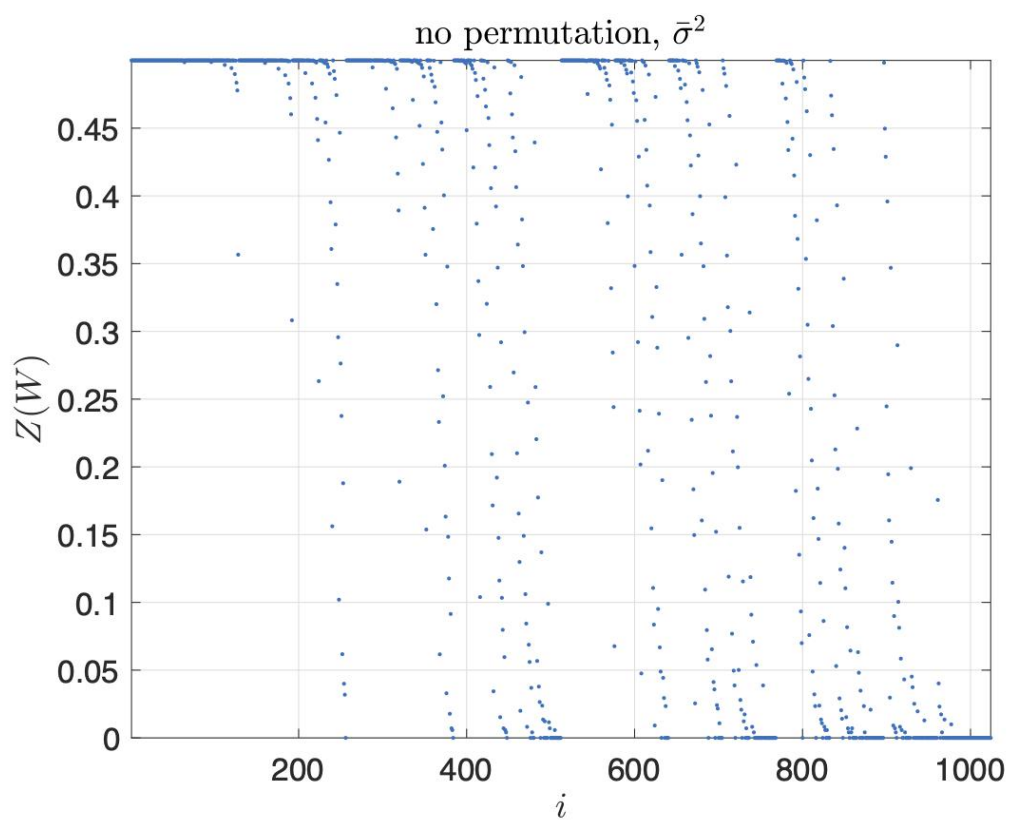
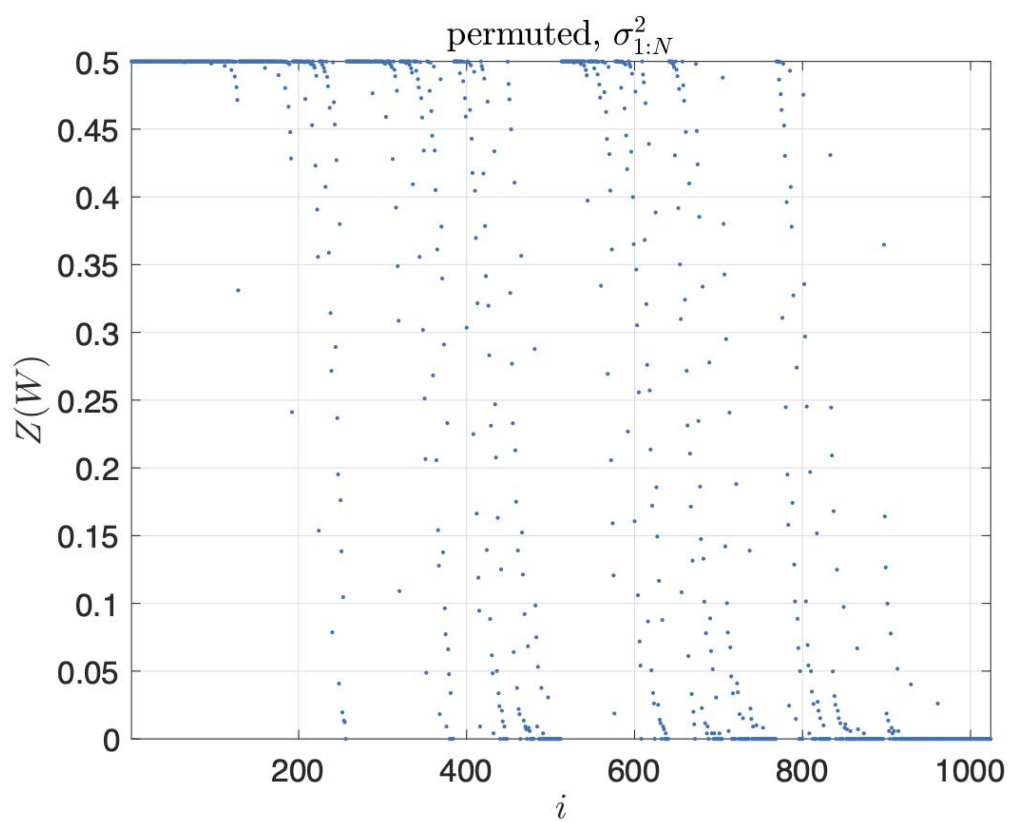
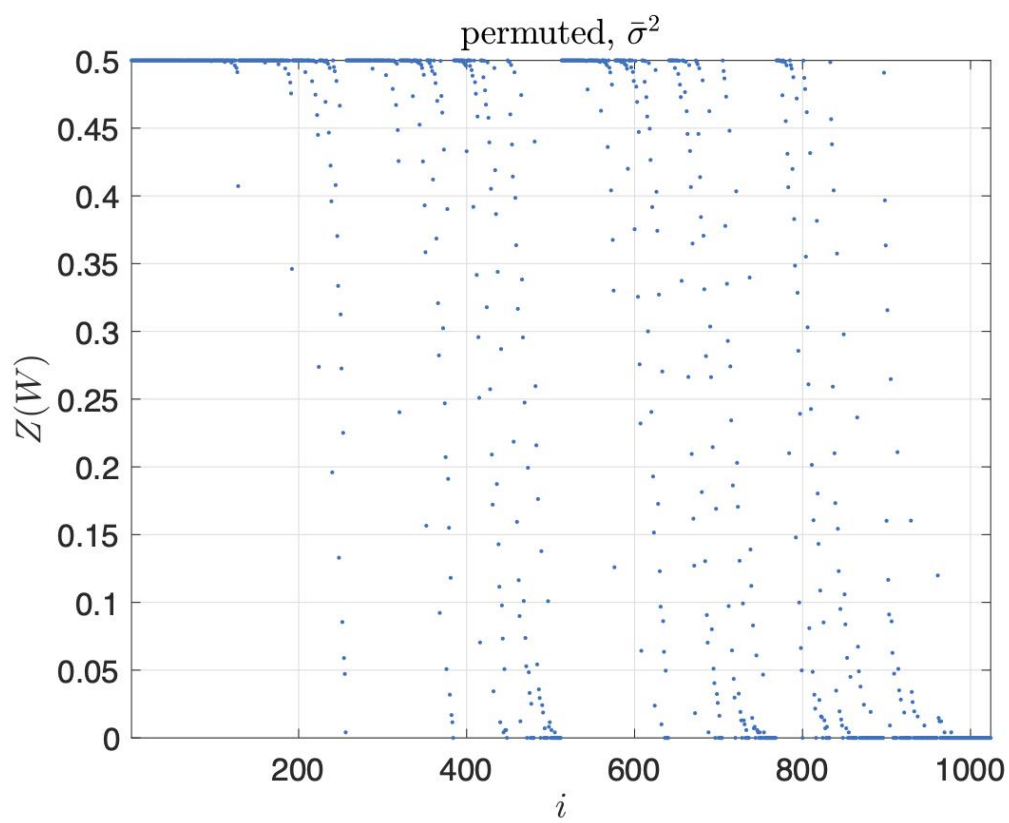


Examples of Polarization of Piecewise-Stationary Channel

I consider a channel with 2-state Poisson Markov memory, i.e., the channel switches between good and bad states. I seed x-nodes with either $\sigma_{1:N}^2$ (the local state at each instant) or $\bar{\sigma}^2$ (the global state). I also tried channel permutation. So there are four combinations. I then run the SC algorithm to count the number of correct trials of each virtual sub-channel. It can be found that in all combinations, the virtual sub-channels are polarized. But if x-nodes are seeded with $\sigma_{1:N}^2$ (the local state at each instant) rather than $\bar{\sigma}^2$ (the global state), the polarization will be better. In addition, channel permutation will also bring polarization gain.





Introduction to Codec Software Package

In this package, there are ten **.m** files and one **.txt** file. The **Polar_reliability_5G.txt** file is obtained from “NR multiplexing and channel coding,” document TSG RAN TS 138.212, 3GPP.

The **.m** files are explained as below.

- The entry file is **test.m**.
- The **algsww.m**, **algww.m**, and **alglw.m** files implement the SWSCAN, WWSCAN, and LWSCAN algorithms, respectively.
- The **polar_enc.m** file implements the encoder of polar codes.
- The **polar_dec_sc.m** file implements the SC decoder of polar codes.
- The **polar_dec_bp.m**, **polar_dec_fp.m**, **channel_upgrade.m**, and **channel_degrade.m** files implement the SCAN decoder of polar codes.

For example, if we want to test 5-state HMM-AWGN channel model with transition probability $1/64$ and global noise variance 0.5 when the decoder is seeded with $1*0.5$, then we can use the following command:

```
test('awgn', [0.5, 1], 'hmm', [5, 1/64], [50, 1e4]);
```

On each CPU core, the loop will be terminated if more than 50 erroneous frames are detected or $1e4$ blocks are tested.

Another example, if we want to test sin-AWGN channel model with local noise variance $0.5(1 + 1*\sin(2\pi*1*i/N))$ when the decoder is seeded with $1*0.5$, we can use the following command:

```
test('awgn', [0.5, 1], 'sin', [1, 1], [50, 1e4]);
```

Running results will be displayed in command window. Each line corresponds to one algorithm, including four terms:

average-running-time, frame-error-rate, bit-error-rate, false-positivity-rate