TURBO AUTOENCODER: DEEP LEARNING BASED CHANNEL CODE

YIHAN JIANG[1], HYEJI KIM[2], HIMANSHU ASNANI[3], SEWOONG OH[4], SREERAM KANNAN[1], PRAMOD VISWANATH[5]

[1]Department of ECE, University of Washington, [2]Samsung AI Center, Cambridge, [3]School of Technology and Computer Science (STCS) Tata Institute of Fundamental Research (TIFR), Mumbai, [4]Allen School of Computer Science & Engineering, University of Washington, [5]Coordinated Science Laboratory, University of Illinois, Urbana-Champaign

PROBLEM SETUP

CHANNEL CODING

- Over-complete autoencoder, with noisy channel in the middle.
- Most widely used channel model, AWGN: $y = x + z, z \sim N(0, \sigma)$.
- The output of encoder must satisfy power constraint.

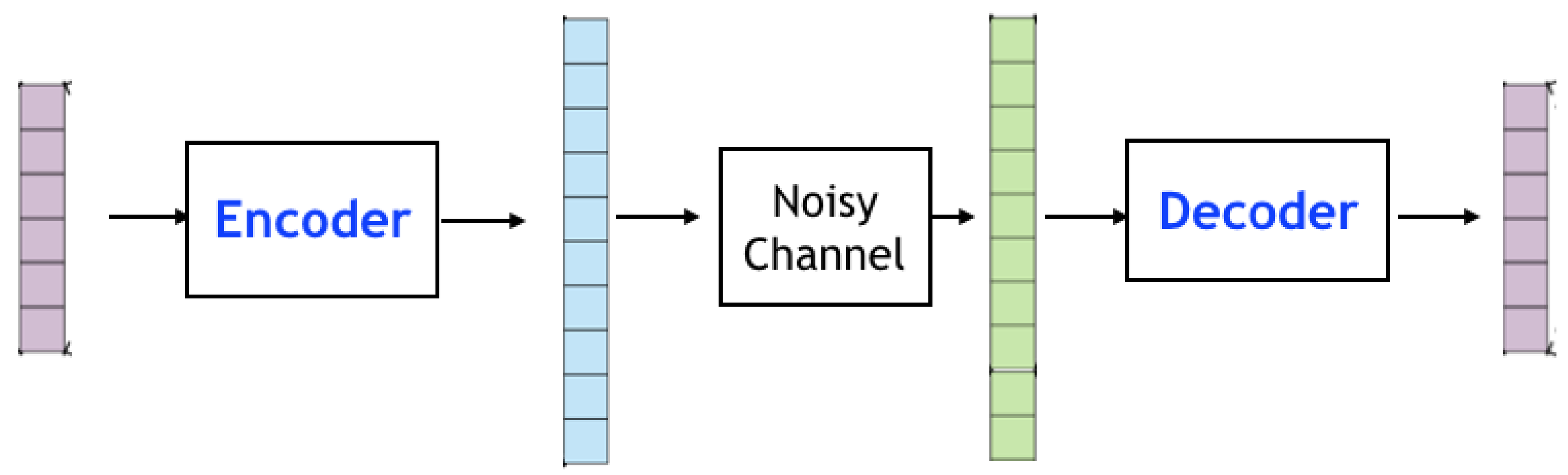

Figure 1: Channel Coding Problem

HUMAN INGENUITY DESIGNED CHANNEL CODES

- Progress in coding theory is sporadic breakthroughs over the past century.
- Turbo, LDPC, and Polar codes have been capacity-approaching on AWGN.

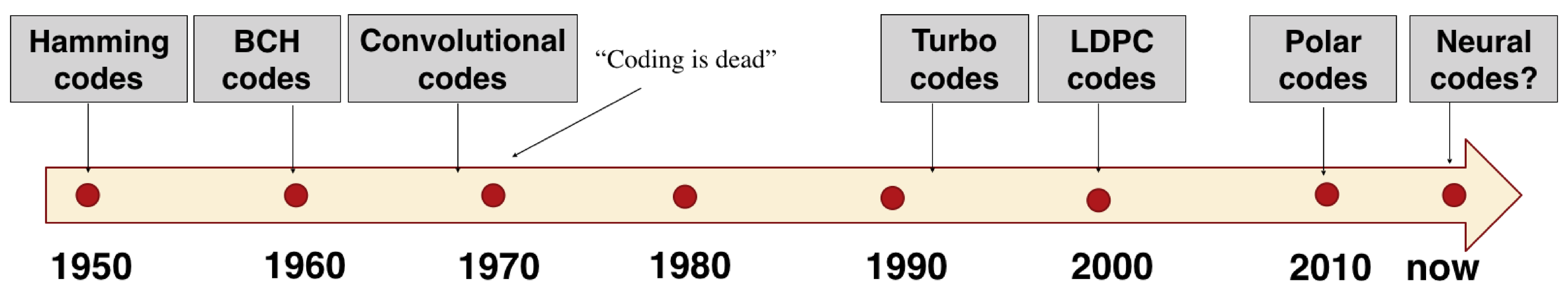

Figure 2: Research Timeline of Channel Codes

CHALLENGES

- Reliability can be further improved for short block lengths.
- Robustness and adaptivity for non-AWGN channels.
- No block length gain with learnable general purpose network.
- Automate the search of codes by solely deep learning.

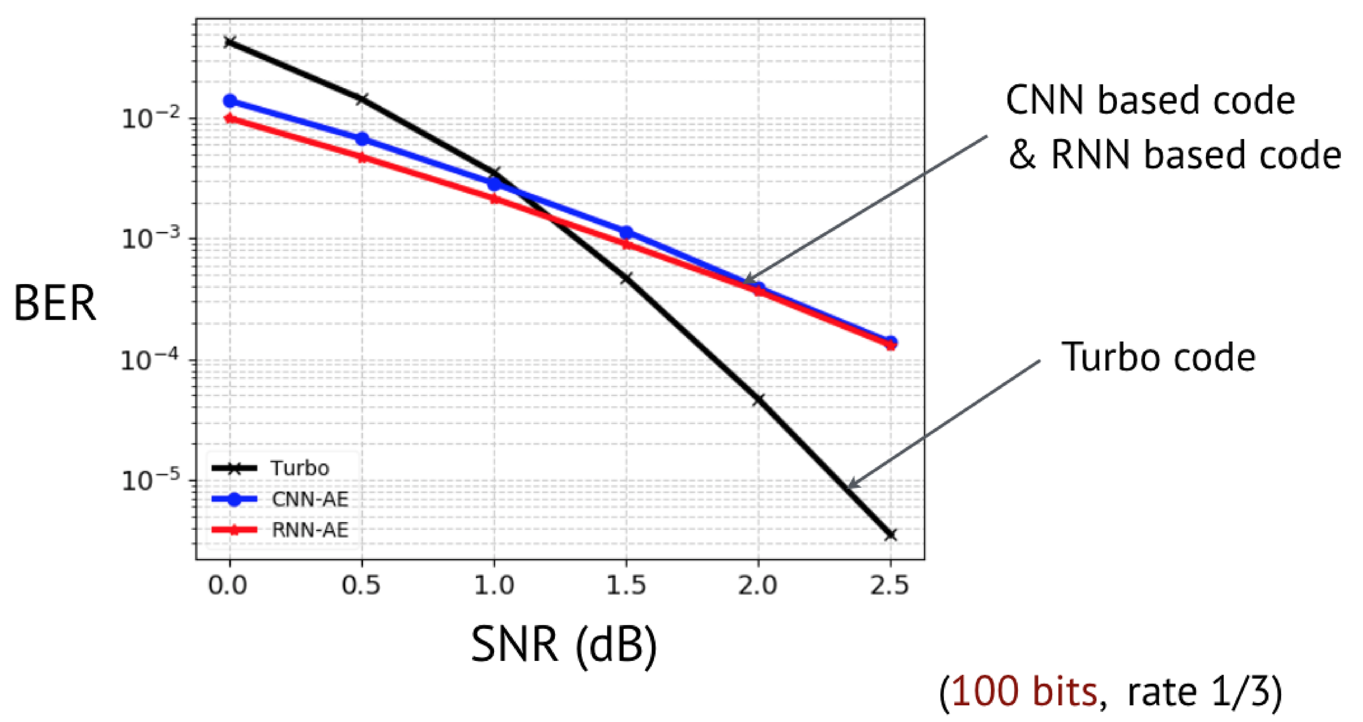

Figure 3: Challenges of using general purpose network for channel coding.

TURBO AUTOENCODER

- Turbo Principle: introduce long term memory with interleaving on encoder.

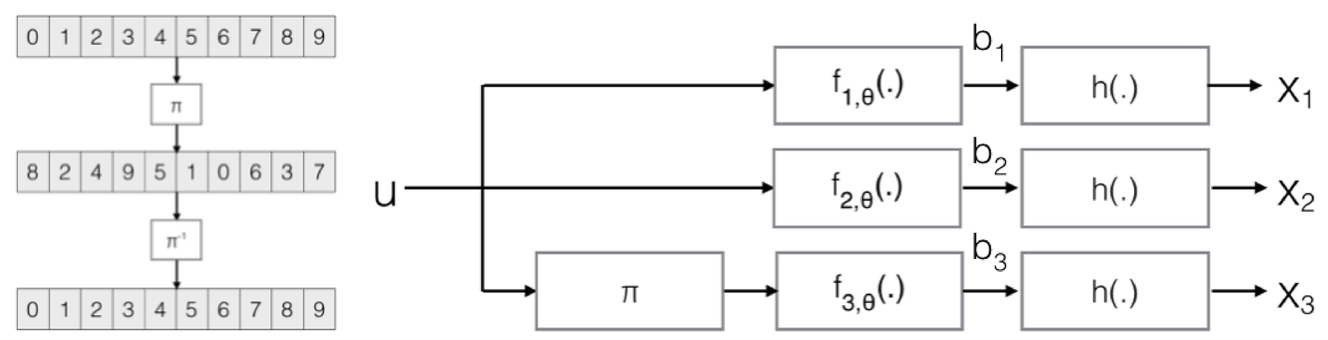

Figure 4: Interleaver (π) and De-Interleaver (π^{-1}) (left) and TurboAE encoder (right).

- Iterative learnable decoding via Belief Propagation.

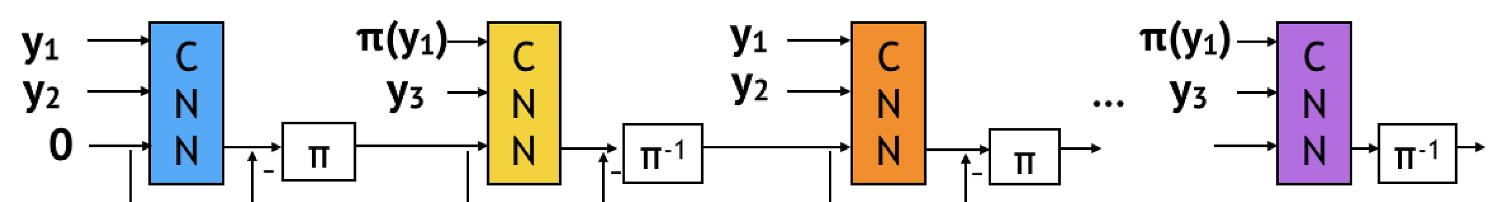

Figure 5: Iterative Decoding, with vector feature representation.

- Training Algorithms: Alternative Training, Large batch size, optimal training SNR.
  - Training SNR: encoder (same as test), decoder (close to Shannon Limit).

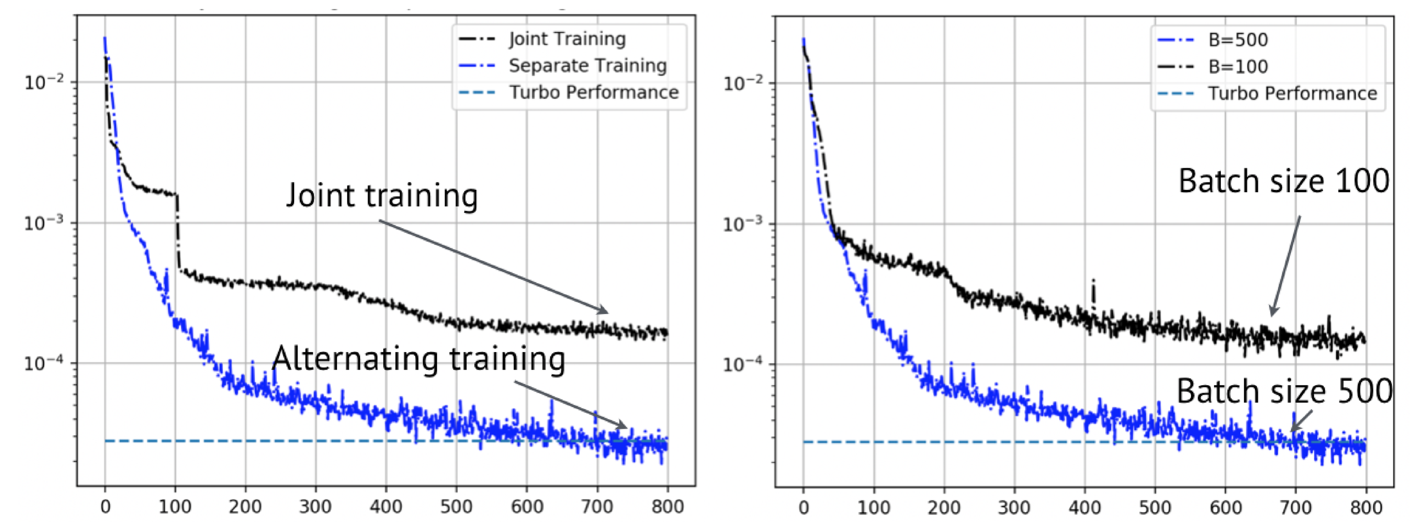

Figure 6: Alternative Training shows smooth learning curve (left), and use large batch size improves performance (right).

- Differentiability via Straight-Through-Estimator (STE) to binarize code.
  - STE forward: $z = sign(x)$, backward-gradient: $\frac{\partial x}{\partial z} = 1_{(|x| \le 1)}$.

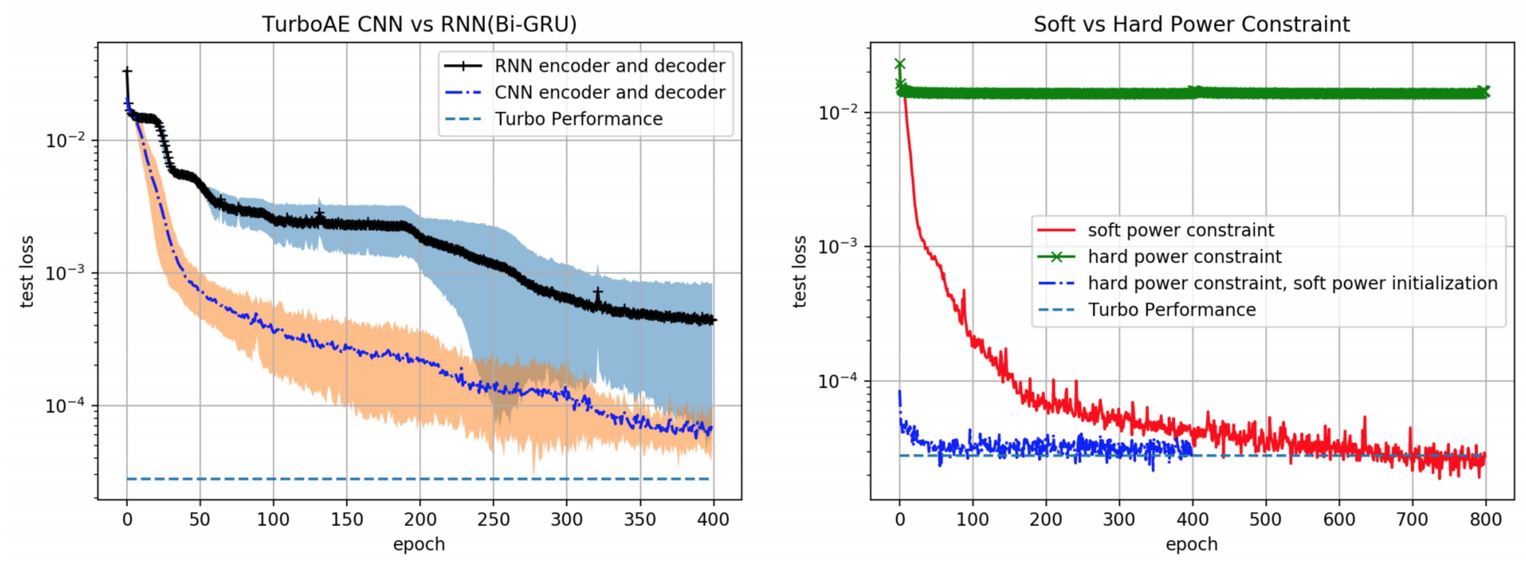

Figure 7: CNN vs RNN in training (left), and STE result.

TURBOAE PERFORMANCE

- Comparable performance to canonical codes in AWGN channel.

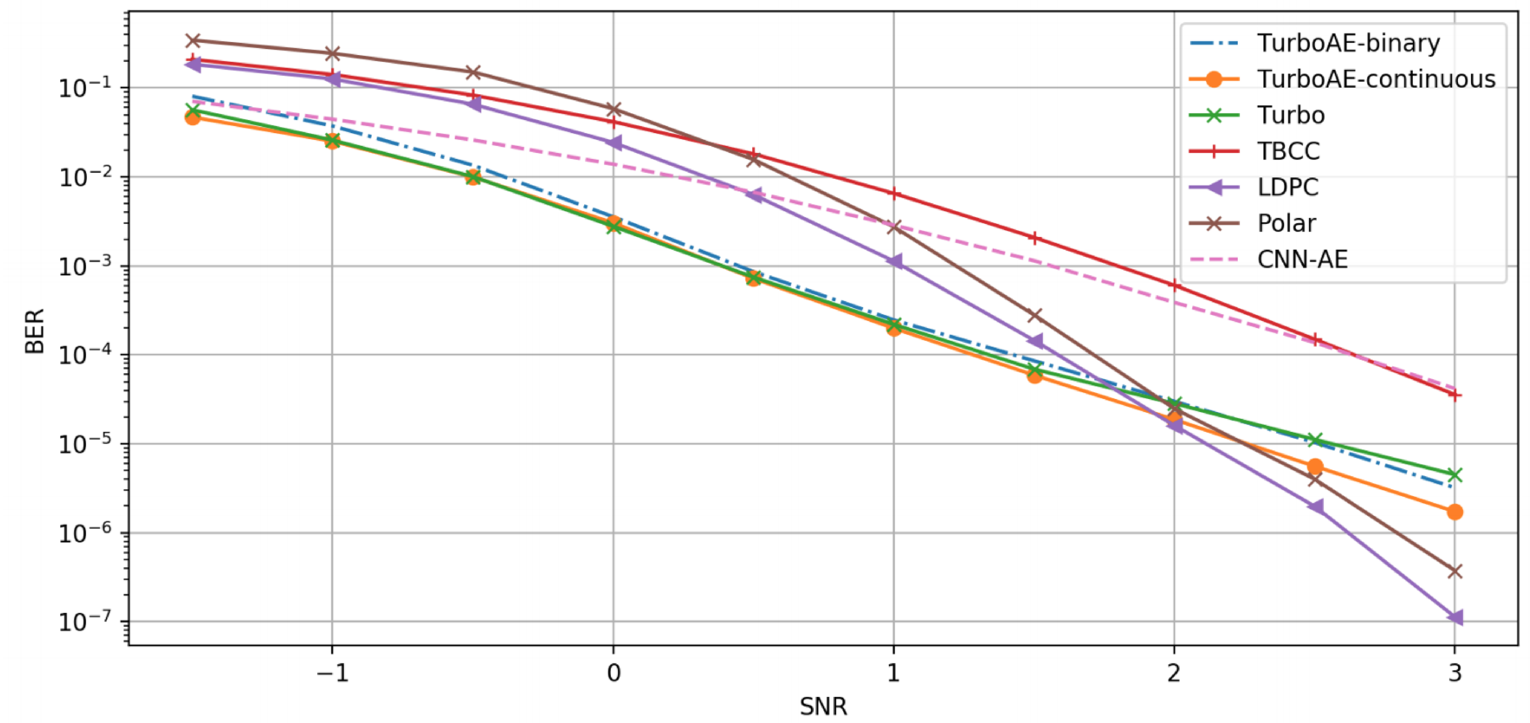

Figure 8: TurboAE vs other canonical codes under block length 100.

- TurboAE reveals coding gain in block length.
- Binarization via STE shows minimal performance drop.

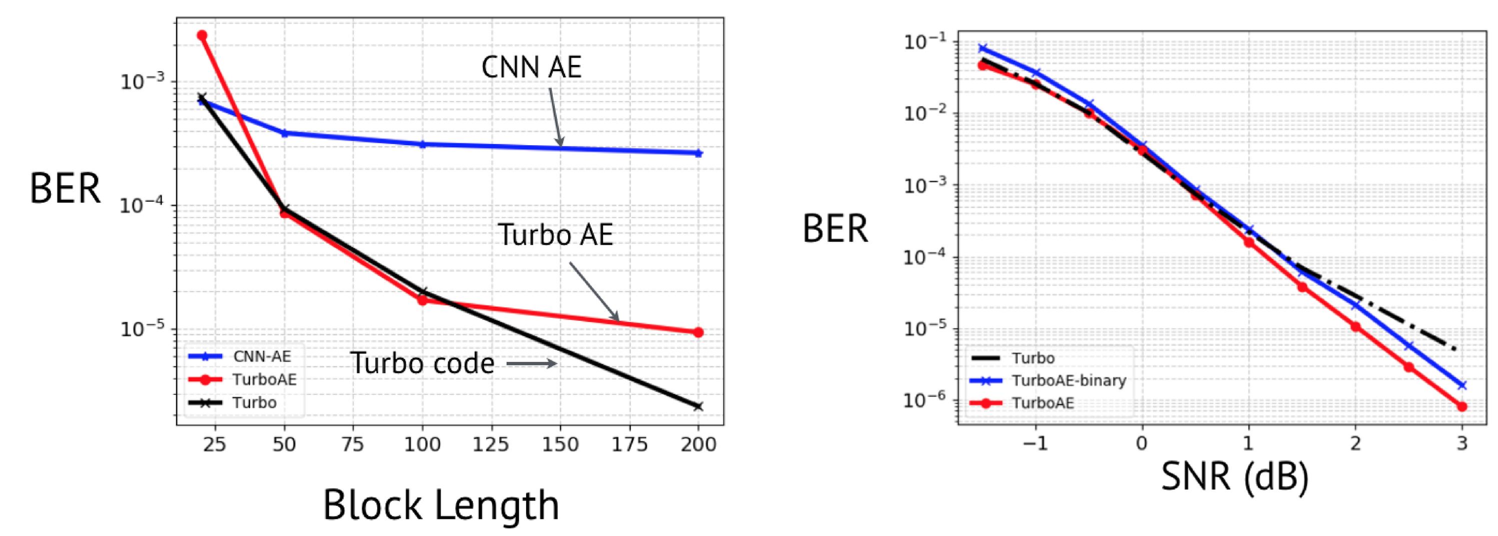

Figure 9: Block Lenght Gain of TurboAE (left), and TurboAE-binary result (right).

- TurboAE is not a large model, even without further model compression.

Metric	CNN encoder	CNN decoder	GRU encoder	GRU decoder	Turbo encoder	Turbo decoder
FLOP/EMO	1.8M	294.15M	334.4M	6.7G	104k	408k
Parameters	152.4k	2.45M	1.14M	2.714M	N/A	N/A

Figure 10: FLOP comparison between TurboAE-RNN, TurboAE-CNN, and Turbo.

- TurboAE outperforms on non-AWGN channels (shown in our paper.)

SUMMARY

- TurboAE offers an alternative method to design channel code: **Key enabler: novel neural architecture and training algorithm design** can lead to good channel code.
- Channel code can be designed **solely** via deep learning.
- Follow-up Directions: (1) Optimizing BLER, (2) high SNR training, (3) very long block length, (4) combining with feedback channel, and (5) joint modulation and channel coding scheme.
- Code repo: https://github.com/yihanjiang/turboae