We thank the reviewers for their insightful comments. Please find the detailed responses below.

Response to Reviewer 1.

Complexity analysis: TurboAE has a linear complexity in the block length, in runtime and computation. We will provide the exact comparison in the revised version. The run time comparison is non-trivial because TurboAE can be run on GPUs and AI-chips which can highly parallelize the computations. We will provide timing experiments for TurboAE and traditional turbo codes run on both GPU and CPU in the revision.

Response to Reviewer 2.

1. Decoder design for non-AWGN: We agree with the reviewer that the statement is confusing and misleading. We just meant to point out that there are heuristics used in practice that leaves room for improvement. We will revise the sentences to make it clear.

2. Comparison to polar and LDPC codes: We will add more figures for comparisons in the revised version. To mention the results briefly,
2-1. For continuous non-AWGN (ATN) channels, the reliability of polar and LDPC codes are very similar to turbo code; TurboAE outperforms polar and LDPC codes by a large margin.
2-2. For binary input channels, TurboAE is comparable to turbo, polar, and LDPC for BSC and is slightly worse than these traditional codes under BEC (Figure 10, Supplementary 3.3). While TurboAE does not outperform traditional codes, achieving a comparable reliability under *discrete* channels requires a major breakthrough. This is because learning an autoencoder including a non-differentiable layer (binarization of code) makes training challenging. We resolve this issue via Straight-Through Estimator.

3. Generalization to longer block lengths: As noted, it is a common issue that deep learning techniques tend not to scale well with the block length. Up until now, deep learning based codes were shown to meet state-of-the-art for AWGN channel for very short block length. By applying the interleaving idea, we can achieve state-of-the-art reliability for block length 100. “Can we go beyond this?” is certainly a central question. A short answer is “it is not easy.” We show the result of direct generalization to longer block lengths (without re-training) in Figure 5 left; as block length increases, BER decreases slowly. However, we want to emphasize that we are the first to achieve such block length gain (error rate decreasing with increasing block lengths) with a neural networks based encoder. And this is achieved without re-training at the larger block-length.
How to increase this block-length gain is an interesting future research direction. Re-training at the larger block-length is slow, but the reliability is improved (Appendix Section 3.2. Figure 8). However, it is still not comparable to traditional turbo codes at that larger block length.

4. Comparison to normal approximation and other codes (Liva et al.): TurboAE meets normal approximation at low SNR (< 2dB) but is worse than normal approximation (and other codes) at high SNRs (> 2dB); we will include the comparison to Liva et al.'s baselines in the revised version. This pattern matches with the pattern in Figure 1, which shows that TurboAE's performance on high SNR is worse than LDPC and Polar codes. We conjecture that it is because, for high SNR regimes, almost all examples shown in the training are easy to label, as they have small noise. We see (relatively) less examples at the decision boundaries, making it hard to train an accurate decoder. Overcoming this challenge is left as a future work. For example, an adversarial training can be used.

5. Comparison to RNN-decoders: We compare TurboAE with DeepTurbo (which is RNN-based decoder) as it achieves state-of-the-art reliability in decoding turbo codes; DeepTurbo outperforms both traditional decoder and CNN-based decoders [26]. In Figure 6 (Section 4.2), we show that TurboAE outperforms DeepTurbo for ATN and AWGN channels with memory. For AWGN channel, TurboAE is comparable to DeepTurbo. We will include this comparison in the modified version.

Response to Reviewer 4.

Advantage of TurboAE over existing codes: It is true that under AWGN channels, TurboAE achieves reliability comparable to the state-of-the-art. We would like to reiterate (1) why this is interesting, and (2) advantages of TurboAE over existing codes:

1. For AWGN channels, existing codes are already very close to optimal, but it took 50 years of mathematicians' efforts to derive these codes. So the central question is not necessarily whether we can beat these existing codes on AWGN, but rather whether we can discover completely new codes, automatically from data. The major contribution of TurboAE is to demonstrate an alternative neural network based approach to discover new codes in a data-driven manner.

2. For several channels beyond AWGN, existing codes can be far from optimal. We demonstrate that for those channels (e.g., ATN, AWGN with memory), TurboAE outperforms existing codes by a large margin (e.g., 5–10x in reliability). Hence, our work defines a new family of possibilities in code and decoder design for open problems in coding theory.