We thank the reviewers for their thoughtful comments and suggestions. Below, we address the reviewers' main concerns and recommendations; our responses will be incorporated in the final version of the paper.

Relevance of the work to NeurIPS [R1, R3]: Our work is focused on a channel coding setting, and the realization of our cluster optimization scheme is geared specifically toward decoder scheduling of sparse graph-based codes. Note that channel coding applications have been of interest to the NeurIPS community previously (e.g. [1, 2, 3, 4] recently). On a high level, as Bayesian inference over graphical models is at the core of many machine learning applications, we believe that learned scheduling of belief propagation (BP) may be similarly applied to BP-based message passing over a factor graph defined by an underlying probabilistic model, making our approach very relevant to NeurIPS.

Providing more intuition [R1, R2]: We note here that our algorithm is dynamic, and depends both on the graph structure and on received channel values: thus, the schedule realization may change for subsequent transmissions, and will outperform a schedule that is fixed in advance. NS relies on the intuition that in loopy BP, the higher the residual of a CN, the further that portion of the graph is from convergence. Hence, scheduling CNs with higher residuals is expected to lead to faster decoder convergence. As NS follows a fixed greedy schedule, there exists a non-zero probability that initially correct, but unreliable, bits are wrongly corrected into an error that is propagated in subsequent iterations. In contrast, our proposed scheme based on Q-learning allows some room for exploration (not just exploitation, as in NS) by scheduling the CN with the highest expected *long-term* residual, mitigating such a potential error propagation.

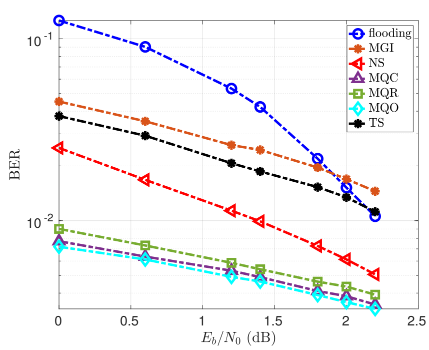
Updates to simulation results [R2, R4]: We have added results for higher SNR (see Fig. (a)) for $(3, 6)$-regular LDPC codes, which indicate that the MQO scheme significantly outperforms the non-RL decoding schemes. The results for $(3, 7)$ AB codes are given in Fig. (b), showing similar results. We also simulated a $(63, 51)$ BCH code. However, since this is a high density parity check code, the training complexity is higher than for LDPC codes. As a snapshot, we obtained a result for an SNR of 3.5 dB with a BER of 10^{-2} for MQO and of $1.3 \cdot 10^{-2}$ for the hypernetwork decoder of [4], showing again the gain achieved by our RL approach. We have also adjusted the metrics of the paper from BEP to BER, as suggested by R4.

Complexity comments [R2, R3]: We note that the time complexity for selecting a CN in line 12 of Alg.1 grows linearly with the total number of CNs, as opposed to being zero for BP flooding. This overhead will be discussed more explicitly in the final paper. The concerns of R3 regarding the comparatively larger state space of longer block lengths, even with clustering, is an area of ongoing work. However, we believe that the substantial gains of our optimized clustering method demonstrated at lower block lengths are promising, and remain an important contribution in and of themselves. As a first approach to mitigating complexity at longer block lengths, we have implemented a Thompson sampling (TS) approach (see Fig. (a)). Another approach is to approximate the Q-table via a neural network.

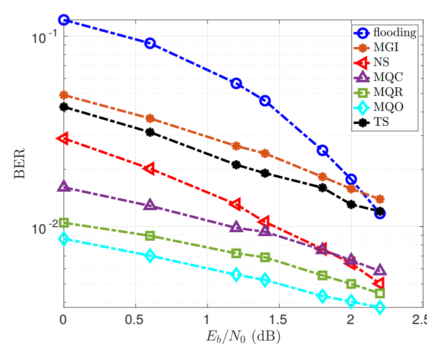
Distinction from previous work [R1]: We thank R1 for the valuable suggestion. Our work also differs from the vast majority of works (including those cited below) in that our decoder is not based on deep learning.

Alternative RL algorithms [R3]: As a first step towards other RL approaches, we have implemented a decoder based on TS (see Fig. (a)), which performs better than flooding, but worse than our proposed Q-learning scheme. In this TS-based scheme we track the densities of the messages via a Gaussian approximation and use the MSE $(m'_{a\to v} - m_{a\to v})^2$ as a non-central chi-square distributed reward, sampled in each sequential decoding step.

(a)



(b)



References

[1] Rishidev Chaudhuri and Ila Fiete. Bipartite expander Hopfield networks as self-decoding high-capacity error correcting codes. In *Advances in Neural Information Processing Systems*, pages 7688–7699, 2019.

[2] Yihan Jiang, Hyeji Kim, Himanshu Asnani, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. Turbo autoencoder: Deep learning based channel codes for point-to-point communication channels. In *Advances in Neural Information Processing Systems*, pages 2758–2768, 2019.

[3] Hyeji Kim, Yihan Jiang, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. Deepcode: Feedback codes via deep learning. In *Advances in Neural Information Processing Systems*, pages 9436–9446, 2018.

[4] Eliya Nachmani and Lior Wolf. Hyper-graph-network decoders for block codes. In *Advances in Neural Information Processing Systems*, pages 2329–2339, 2019.