We thank all reviewers for their time and thoughtful comments. **Summary of positive reviews:** Reviewers liked our proposed algorithm (BPNN) because it is innovative (@R1), elegant (@R3), interesting (@R4), and theoretically grounded (@R2, @R4) with clear proofs (@R3). Our approach shows solid experimental improvements for computing the partition function of Ising models, community detection problems, and factor graphs representing boolean formulae (@R2, @R3, @R1). Additionally, we demonstrate robustness to perturbations in the test distribution (@R1). Reviewers thought that our paper was generally well written (@R2) and that it does a good job of explaining our contributions and their significance (@R3), although it is inaccessible to readers who aren't familiar with the variational view of BP (@R3). **Summary of negative reviews:** The most serious concerns were that it was challenging to understand our model's structure/output (@R4) and the training protocol we used (@R1). Additionally, there were questions/confusions about our experimental setups.

@R1, @R4: What is the output of BPNN? BPNNs consist of two parts. First, iterative BPNN layers output messages, analogous to standard BP. These messages are used to compute beliefs using the same equations as for BP. Second, the beliefs are passed into a Bethe free energy layer (BPNN-B) that outputs an estimate of the log partition function. This layer generalizes the Bethe approximation by performing regression from beliefs to $\ln(Z)$. Alternatively, when the standard Bethe approximation is used in place of the (trainable) BPNN-B layer, BPNN provides additional guarantees.

@R1: What is the training protocol? In all our experiments, we initialized the BPNN to output the Bethe approximation obtained by running BP for a fixed number of iterations. We used the mean squared error between the BPNN prediction and the ground truth log partition function as our training loss. When training BPNN-D without a BPNN-B layer, we ran BPNN-D for a random number of iterations between 5 and 30 at every training step to encourage quick convergence to a fixed point. (When training with a BPNN-B layer, iterative layers were applied for a constant number of iterations, as convergence to a fixed point is no longer required.) We will highlight these details in the final copy.

@R4: When is convergence to a fixed point unnecessary? How was the BPNN-B layer designed? We can think of BPNN as a trainable computation graph (neural network) that mimics the standard (unrolled) BP computation. Convergence to a fixed point is unnecessary for good predictive accuracy; however, this extra flexibility makes theoretical analysis more difficult. The Bethe layer (BPNN-B) was carefully designed to generalize the Bethe approximation while maintaining invariance to factor graph isomorphism (please see Lemma 4 in the appendix).

@R1, @R4: What version of BP is compared against? What does "standard BP" refer to? We compared against BP where we tuned the traditional damping coefficient and message update strategy (sequential or parallel) for best results. "Standard BP" refers to belief propagation run with traditional damping but without learned modifications to messages (via the operator $H(\cdot)$).

@R4: Did you compare with a strong, baseline algorithm that utilized the same training dataset as BPNN? Yes, we compared with and found significant improvements over the Graph Isomorphism Network GNN.

@R3, @R4: Did your experiments include problems where BP converges well? Yes, BP converged on all community detection problems and 88% of Ising models (line 178).

@R1: "Why would we expect BPNN-D to converge to better local optima than BP? Because the learned H function helps avoid local optima? If this is true, it is an interesting observation by itself." Yes, that is the intution and we agree it is interesting! Fixed points of BP are local optima of the Bethe free energy. By training BPNN-D to predict the exact partition function, BPNN-D can learn to find better local optimum.

@R1: "I'm very curious how overfit (if at all) these models are to the domain on which they are trained." We explored cross domain generalization in our propositional model counting experiments and found that the quality of results is dependent on the similarity of domains. We found that we could learn improvements that translate across a broad variety of domains, although the improvements were less dramatic (lines 289-293).

@R4: "The theorems in the paper ... do not add anything in understanding when BP on loopy graphs converge" That is correct, our theorems make no claims about the convergence properties of BP on loopy graphs. Our theorems are given to *precisely characterize the relationship between fixed points of BP and BPNN-D* in terms of properties of the learned operator $H(\cdot)$.

@R4: "The theorems in the paper are essentially trivial ... Without going through all provided proofs in detail, the theoretical justification of the stated theorems seems to be correct." Even though the theoretical results may seem intuitively correct in hindsight, they provide valuable insights into BPNN. We think it is unfair to characterize the theorems as trivial because similar results do not hold for standard GNNs.

@R3, @R4: Can BPNN be used to estimate marginals? Yes. We focused on estimating the partition function because these estimates are better understood for BP. However, in a small set of experiments on estimating marginals for community detection, we found large improvements over BP and GNNs. This is a promising direction for future work.

@R3: Can the BPNN-B layer be modified for computational efficiency on high dimensional factors? Yes. The BPNN-B layer can be modified to use an attention mechanism for efficient invariance given high dimensional factors.

@R2: Will you open source your code after publication? Yes!