Thank you all for the helpful reviews. We appreciate the acknowledgment that our paper is "well written and clearly motivated" and "tackles an important problem in graphical model optimization". We're happy to hear that we have shown an "interesting idea" and "clever contribution" that achieves "state-of-the-art performance" and is "easily reproducible".

We start by addressing Reviewer 4's concern that "calculating the exact bound for selective-SPNs" is not a "contribution of this paper". Thank you for pointing out the relevant work by Lowd and Domingos (2010). Our problem setup is indeed similar. However, our method of computing the ELBO gradient is in fact **novel** and is much **more efficient** than their method, on two fronts.

1. First, using their notation, their computation of all gradients takes $O((n+m)e)$, whereas our computation of all gradients takes $O(ne)$, since $n \to t$ and $e \to kn$ in our notation (line 226 in our paper). Since they use small circuits, they claim $m < n$ and $O((n+m)e) = O(ne)$. However, circuits in recent years have grown to hundreds of thousands in size (and in our experiments $m \gg n$), so $O(ne)$ **is much better than** $O(me)$.
2. Second, and equally important, is that we compute all the gradients in one pass of the circuit thanks to backpropagation, which allows for **GPU optimization**. For large models, we saw a 5-10x speedup with GPUs. Their method computes the gradient of each individual parameter separately (cf. Eq5 in their paper) and requires fixing different parts of the circuit constant, which cannot be easily optimized with GPUs.

As such, our method is much faster both in theory and in practice, and is one of the main contributions of this paper.

TRWBP: Thanks for the suggestion of comparing against tree-reweighted BP. We use TRWBP from libDAI, using 10000 random spanning tree samples. For Fig2, TRWBP does better than SPN-VI for 4x4 but worse for 8x8, 16x16, and 32x32. For Tab1, TRWBP is generally worse than LBP. For example (will include everything in final version):

DBN11	DBN13	DBN15	grid10	Grids11	Grids13	Grids15	relat.	Seg11	Seg13	Seg15
319.33	406.20	352.15	908.14	487.16	965.39	800.23	746.14	-44.94	-67.38	-58.44

SMF-VI: We could not find a competitive implementation of SMF-VI that uses GPUs. We had to implement our own version of SMF-VI that is GPU-optimized (see the supplementary zip file), but it only works for degree-2 interactions (as shown in Fig2 for Ising models). We were not able to easily extend our GPU-optimized impl. to general graphical models for Tab1, and comparing against a non-GPU impl. seems unfair (but happy to include if reviewers want).

R1 "leave out tree reweighted BP": Thanks, please see **TRWBP**.

"proofs of the results", "generated structured is selective": Ok, we will clarify this. Yes, the main intuition is to generate partitions of the support using partitions of the children variables, in a decomposition-like approach.

"Def 1" disjoint scopes: Good point, we will reword this to say scope.

"limited to binary": The circuits can be over categorical variables as well. We focused on binary variables for simplicity.

"(page 4, line 164) monomial": Ok, we will clarify that there is a constant in our setting.

"citation is not in the proper format": Thanks, we will fix the formatting.

"better to assume a representation in terms of factors": Ok we will keep this consistent.

"moments in structured decomposable probabilistic circuits": Yes, we did comment on this at the end of Related Work.

"prove Theorem 1 directly for the derivatives": Good point, we will consider this.

"What is a selective mixture?": This refers to shallow SPNs of depth 2.

R2 "general discrete": Sorry, will clarify that our framework can be extended to general discrete settings by using discrete leaf distributions, but our experiments currently are only binary.

"Why did you not compare to SMF-VI": Please see **SMF-VI**.

R3 "full expressiveness": Yes, you understood correctly. We will clarify that the family of models is fully expressive but the circuit we construct is an approximation.

"detailed explanation...inference pipeline": Ok we can include more detail. Briefly: we backprop on the exact ELBO to get gradients w.r.t. SPN params, and perform gradient steps to optimize the lower bound estimate of partition function.

"WMC based inference": Should be possible to convert WMC to an equivalent PGM problem and apply our method.

R4 "limited to the models without hidden variables": For models with hidden variables, our method should be able to handle $\sum_z p(x,z)$ as long as $p(x,z)$ has the right structure for every x.

"studied before by Lowd and Domingos": Thanks for the reference. Please see above (ours is much more efficient).

"tree-reweighted belief propagation": Please see **TRWBP**.

"neural variational inference": Thanks for the reference. Unfortunately their repository did not include their code for Ising models. Nevertheless, their paper clarifies that the upper bound "will not be directly applicable to highly peaked and multimodal distributions...such as an Ising model". So in their experiments they only bound the partition function of an **approximation** of the original Ising model, and only scale to size 5x5 (we scale to 32x32).

"Inference network discussed in Wiseman...": Thanks again! We will discuss this too in the related work.

"SMF-VI should also be in Table 1": Please see **SMF-VI**.