We thank the reviewers for the constructive feedback and detailed comments which we integrate in the final version.

R1/R2/R3: *"comparison with [7] for boosted decision trees and to neural networks"*
At submission time no code was available for [7] which is why we did not compare to them. Please note that we optimize directly an upper bound on the adversarial loss whereas this is only approximately true for [7]. In Table 1 we compare our provably robust boosted trees to [7] (same setting for [7] as ours: we fit boosted depth 4 trees with 80% of the training data and use the rest as validation set for model selection). For [7] we use the exact robust test error (RTE) [22] for model selection, whereas for us we use our upper bound on RTE (URTE). For [7] we use a coarser grid for large number of iterations as RTE is expensive to evaluate. We see that our URTE is for 6 out of 7 datasets smaller than their RTE sometimes with large margin e.g. on diabetes. Our better URTE comes at the price of worse test error but this is a well-known phenomena for neural networks, that methods enforcing better RTE suffer in test error. Our LRTE values improved as we have come up with a new attack scheme - now LRTE and URTE are tight.

Table 1: Comparison of the boosted trees of [7] to the results of our boosted trees reported in the paper. The shown time is for boosted trees of [7] the computation of the RTE for the final model with the MILP of [22] (adapted to a feasibility problem for existence of an adv. example within l_∞-ball) and for URTE with our algorithm. All numbers are for the full test set.

		Chen et al [7], depth=4				Our provably robust trees, depth=4				
Dataset	l_∞ ϵ	TE	RTE	# trees	Time RTE [22]	TE	LRTE	URTE	# trees	Time URTE (ours)
breast-cancer	0.3	0.7	13.1	8	5.3s	2.9	10.2	**10.2**	1	**0.1ms**
diabetes	0.05	22.1	40.3	5	4.0s	28.6	33.1	**33.1**	3	**0.8ms**
cod-rna	0.025	10.2	24.2	20	1.4h	8.3	23.2	**23.2**	14	**0.5s**
MNIST 2-6	0.3	0.5	6.9	1000	2.0m	0.7	4.8	**5.0**	47	**0.2s**
FMNIST shoes	0.1	3.1	13.2	20	58.3s	4.7	10.5	**10.5**	8	**0.1s**
GTS 100-rw	8/255	1.5	**9.7**	20	35.9s	4.7	10.1	10.1	11	**0.2s**
GTS 30-70	8/255	11.5	28.8	20	23.4s	14.9	27.2	**27.2**	14	**0.4s**
MNIST	0.3	2.0	31.2	200	2.7h	4.8	14.6	**18.5**	55	**2.5m**
FMNIST	0.1	14.4	65.1	200	3.8h	15.3	23.5	**25.4**	25	**1.2m**

We extended our approach to multi-class using one-vs-all. We fitted tree ensembles of depth 14. For MNIST with $\epsilon = 0.3$ we get a URTE of 18.5% versus 31.2% for [7]. Our URTE is better than that reported for neural networks (NNs) (33.6% [45], 19.3% [47]) and only the very recent [17] improved this to 8.1%. For FMNIST we get 25.4% URTE vs 65.1% RTE for [7] whereas NNs achieve 30.7% URTE [10] (with 26.6% LRTE) so that our tree ensemble is more robust. **This shows that regarding provable robustness tree ensembles can be competitive with NNs.**

R1/R2/R3: *"comparison to adversarial training (AT)"*
We tried AT as in [22] and obtained much worse robustness than ours. Different from the l_0-attack of [22] for an l_∞-attack all features are perturbed and that leads to suboptimal initial splits from which the ensemble does not recover. We think that AT should not be used if one has a tight and scalable upper bound on the robust loss as AT provides only a lower bound and minimization of an upper bound makes more sense than minimization of a lower bound.

R1: *"c) Approximate upper bounds on robustness of stump and tree ensembles".*
We want to clarify that our upper bound on the adversarial loss is not approximate, but a strict upper bound.

R1: *"in the case of decision stumps, the MILP may very well be solvable quickly".*
The MILP of [22] scales up to larger tree ensembles when changing it to a feasibility problem for the computation of the RTE rather than the minimal adv. perturbation. However, our upper bound computation which is very tight (see Table 1) is about 100x faster. For decision stumps our exact algorithm has runtime complexity $O(nT \log T)$ whereas the MILP has no polynomial runtime guarantees and in practice the MILP is several times slower. We can't compare the running time directly as we need time to transfer our tree ensembles into the code of [7] to access the MILP [22].

R1: *"the theoretical results in the paper are not especially deep, even though they are certainly novel and interesting"*
Scalable provably robust training need not be complicated. IBP [17] the state-of-the-art method for provably robust NNs is based on "simple" interval arithmetics, and theoretically less involved than [32, 44] which are hard to scale.

R1: *"Some more motivation for focusing on decision stumps would be nice."*
For simple data sets boosted stumps are sufficient and more interpretable than boosted trees. In terms of RTE our exact decision stumps outperform our robust tree ensemble on diabetes and cod-rna. Apart from linear models this is up to our knowledge the first scalable, exact algorithm for the minimization of the robust loss. See also [15] for an example where boosted stumps have better generalization properties than boosted trees.

R2: *"Is there a way to bring down the training computational complexity down from $O(n^2)$"*
We investigate if it can be improved to $O(n \log n)$ but have not succeeded yet. One heuristic is to use a small subset of the thresholds for large datasets. Empirically, this yields only small loss in URTE but a significant speed-up of training.

R3: *"... more discussion of interpretability of boosted decision trees/stumps"*
We agree that this is very interesting and will include more analysis along the lines we have done already.