We begin by sincerely thanking all reviewers for carefully reading our paper and providing useful feedback. Below, we respond to all concerns raised in the initial reviews. If accepted, we will amend the manuscript accordingly.

Generalization properties of heterogeneous boosted ensembles (R2). In the paper, we provide extensive *empirical* evidence that such ensembles can improve generalization vs. homogenous boosted decision trees. From the *theoretical* perspective, margin-based generalization bounds for heterogeneous boosted ensembles have been previously derived in the DeepBoost paper (Cortes et al. 2014). Specifically, the authors showed that the generalization error is bounded from above by the empirical margin error plus a weighted sum over the Rademacher complexities of the different base hypothesis subclasses. The weights correspond to how many times each subclass appears in the ensemble. The authors argue that heterogeneous ensembles have nice generalization properties, since by selecting a highly complex subclass fairly *infrequently*, one may be able to reduce the empirical margin error without blowing up the complexity term.

What is the role of Θ (R4)? To gain more intuition, let us consider an alternate but equivalent definition:

$$\Theta = \min_{c \in Range(B)} \mathbb{E}_{k \sim \Phi} \left[\max_{j \in I_k} [\cos(B_j, c)] \right]$$

The term inside the expectation, for a given subclass k, measures the closest fit between a hypothesis in the subclass and the direction c (larger is better). Thus, Θ measures the worst-case *expected* closest fit, where the expectation is taken over the PMF Φ. The result of Theorem 2 tells us that larger values of Θ will lead to faster convergence.

Can we use our theory to determine the best PMF and/or subclasses (R3, R4)? Theorem 2 provides a *guarantee* that HNBM will converge in terms of training loss. While the choice of PMF and/or subclasses affect the convergence rate, it is not necessarily a good idea to explicitly optimize them for fast convergence. To see this, let us imagine there exists a perfectly *dense* subclass with index k^* for which, for all $c \in Range(B)$, there exists a $j' \in I_{k^*}$ such that $\cos(B_{j'}, c) = 1$. In this case, our theoretical results imply that convergence rate can be maximized by assigning all mass in the PMF to subclass k^*. However, a subclass with this property is likely to be very complex, which is not ideal from the perspective of generalization (see above). Thus, it is preferable to parameterize the PMF and use cross-validation.

Which hypothesis classes satisfy Assumption 3 in practice (R2)? The role of Assumption 3 is to separate the *scale* of the hypothesis (i.e., σ) from the *structure* of the hypothesis (i.e., b). This makes the derivations simpler and easier to follow. The only restriction it places on the hypothesis subclasses in practice is that they satisfy a scalability assumption: if b belong to the subclass, then σb must along belong to the subclass for any $\sigma \in \mathbb{R}$. This is certainly true for decision tree regressors, linear regressors and most other regressors we can think of. The argument that the number of (normalized) hypotheses is finite when represented using machine precision is a common argument found in machine learning textbooks (see for example Section 2.3.1 of the textbook by Shalev-Shwartz and Ben-David).

What is the main strength of MixBoost (R2)? The main strength of MixBoost is that it can achieve better generalization than boosting machines that use trees alone (e.g. XGBoost, LightGBM) without significantly increasing the tuning time required, which is a problem suffered by other heterogeneous boosting machines such as KTBoost.

What motivated the choice of base hypothesis subclasses for MixBoost (R3, R4)? Our choices for the base hypothesis subclasses are motivated by existing ideas from the literature: using decision trees of varying depth is inspired by DeepBoost (Cortes et al. 2014) and using LRFs was inspired by KTBoost's use of kernels (Sigrist 2019).

What motivated the choice of the PMF used by MixBoost (R3, R4)? This PMF was motivated by a desire to keep the number of hyper-parameters low (in our case, it depends only on p_t, D_{min} and D_{max}) whilst allowing a high-degree of heterogeneity. We intend to pursue other, more general, choices for the PMF in our future research.

Optimal hyper-parameters values (R4). The optimal values for the parameters D_{min}, D_{max} and p_t for the Kaggle datasets are provided in the table below. We see that while the decision trees used are fairly homogeneous, between 7% and 9% of the hypotheses in the resulting ensembles correspond to LRFs rather than trees.

Dataset	p_t	D_{min}	D_{max}
Credit Card Fraud	0.912	1	1
Rossmann Store Sales	0.925	18	19
Mercari Price Suggestion	0.934	17	17

Results for $p_t = 1$ (R4). The results labelled MIX-T in Figures 1 and 2 correspond exactly to the case where $p_t = 1$, and the results labelled MIX correspond to the case where p_t is tuned.

Why is the maximum number of histogram bins 256 (R2)? We use a single byte for performance reasons.

Could the Fastfood technique (Le et. al 2013) by used instead of random Fourier features (R2)? We were not aware of this work until now but it certainly looks like it could be used to improve the MixBoost implementation further.