We would like to thank all reviewers for their invaluable feedback. The next revision of the paper will include fixes for all typos that were mentioned. Responses for questions raised by each reviewer are below.

Reviewer #1

We would like to clarify questions about the theory you've raised. PolyTree framework represent tree and ensembles not as a sum of weights over **leaves**, but over **right split indicator functions** ($c(x) = I\{x > b\}$, see line 57). This fact is a core idea and should be kept in mind to understand most of the propositions: *A) l74: I do not understand condition 2. A leaf can be bounded from the left and the right, so can't a feature be present twice?* The feature will still be represented only once. Let look at leaf $x \in (0, 1]$. This leaf is product of two indicator function $I\{x > 0\}I\{x \leq 1\}$, this is equal to $I\{x > 0\}(1 - I\{x > 1\} = 1 - I\{x > 0\}I\{x > 1\} = 1 - I\{x > 1\}$. $I\{x > 1\}$ is a stronger condition, thus $I\{x > 0\}$ could be removed. We will add this example to the next revision of the paper to clarify this property. *Theorem 1: Wouldn't it be possible to construct an ensemble H' from an ensemble H by splitting one leaf l and assigning the weigh of the old leaf in the new leaves in H'?* When the ensemble is represented as a sum over leaves, there indeed will be two different ensembles. But, both ensembles will generate the same PolyTree representation, because we perform a reduction of weights for monomials with the same set of conditions (such leaf will generate 2 monomials with the same absolute weights, but different signs, after reduction terms will disappear). This comes from condition 2 (line 74). *l105: In this section, we (...) set up a task of tree shape change in an ensemble. We represent a tree ensemble as a sum of trees of fixed shape. —> unclear to me whether the shape is changed or not.* By fixed shape we mean a restriction on a set of possible tree structures. For example, tree of depth 6; balanced trees; symmetric oblivious trees and so on. *equation 2 seems wrong...* Thank you, this is a typo and will be fixed. *Comments about notations* By 2^C we denote set of all possible monomial structures, and $M \in 2^C$ represent some monomial from this set. We use letter d to denote the depth of decision tree, and in the context of line 85 this is a maximum depth of tree in the ensemble. We'll add all this to paragraph with notation. *Provide code* We have Java-based implementation on GitHub and will include the link with non-anonymous version. We are also working on an implementation of proposed methods as a part of one of the major GBDT libraries and it'll be finished before NeurIPS2019.

Reviewer #2

- *In the last paragraph of section 3, the execution time of an original ensemble with the transformed one using the proposed algorithm is compared; nevertheless, the overhead time added by this transformation is not mentioned.* Our greedy algorithm is quite slow and could take several seconds to complete. Also, this experiment is a proof-of-concept and we did not optimize it to run as fast as possible. Mostly, such transformations are interesting for production tasks where the same model could be used for days. Transformation time could be big, but it is done once, thus overhead is negligible. *...Symmetric trees can highly benefit from model reduction because of having many zero leaves...* For more clarity, there were balanced trees, like on in XGBoost, but symmetric oblivious trees used in CatBoost. Such trees are often not the best choice to learn on one-hot-encoded datasets, so exploring pruning strategy on this type of trees also requires an in-depth study of what type of trees and when we need to use for such data. It is interesting, but there is enough space and we will divert from the main topic — PolyTree framework introduction.

Reviewer #3

Why the level of dependency grows with the number of conditions in M. PolyTree decomposition is similar to n-way ANOVA decomposition, where dependence for factors x, y, z is decomposed to main effects (x, y, z) and their iterations (xy, yz, xyz). In ANOVA x, y, z are categorical factors, while in PolyTree it is right split indicator functions. - *Authors do not mentioned if the cross validation is applied or not. This latter should be applied.* The cross-validation is used during parameter tuning, but the final metric is computed on the fixed independent test set (experiment simulates practical usage of GBDT — choose hyper-parameters / model via cross-validation and then use it in production). *For section 3 relative to theoretical analysis, experiments remain poor, testing on only one set (Higg dataset) is insufficient to make reliable conclusions. More data sets should be tested.* This example shows, that there are certain type of problems, where to get the better model we need to use one type of tree shapes and are able to benefit from the other shape during exploitation. We don't insist that this situation is universal, but just show its existence. Systematic study what tree shape needed for what problem, and what benefits could be achieved for the specific problem at hand, requires space; we are working on this as follow-up work. *Why what is tested in Table 1 is only sets of binary classes. What about sets with multiple classes.* Its a topic for an individual study. There are several ways of how to perform multiclassification in boosting. For some types (like one-vs-all) the problem reduces to binary one, while for others not. There could be several ways to generalized PolyTree for multiclass, and each one should be explored. But this article introduces PolyTree framework and we don't have enough space to explore this problem in details. *In Tabe 1, How to explain that there is no significant improvement In AUC criterion between trained ensembles and pruned ensembles. In fact, you get almost the same AUC for both methods* For the trained ensemble, we have used the early-stopping strategy. GBDT with early stopping usually provided the state-of-the-art model. Thus, this model provides a very strong baseline that is hard to beat with the smaller model.