We would like to thank the reviewers for their helpful comments and their thorough evaluation of our work. We are encouraged that reviewers find our paper clear and well written (R1, R2, R3) and our method to be theoretically sound (R1, R2), of high practical impact (R2, R3), and intuitive and easy to implement (R3). Below, we address the issues that have been raised.

[R1] Should the clusters merge in the next layer?: The output of the attention layer involves (a) residual connections: this means queries in the same cluster can have different values (b) multi-head attention: this allows for a query to be clustered differently in two different heads. This is followed by a feed-forward projection, which results in new clustering patterns in the subsequent layers.

[R1] Is LSH-X a simpler version of the Reformer?: LSH-X refers to Reformer as described in the paper except for the reversible layers. Reversible layers is a technique introduced by Gomez et al. (2017) and is orthogonal and independent to both clustered and LSH attention. As our focus is on the efficiency of the attention implementation, we do not consider the reversible layers for both methods.

[R1] Can Reformer be used for heterogeneous queries and keys?: Using Reformer with heterogeneous queries and keys, requires significant changes in its core component, namely sorting and chunking the hash values to create query/key groups. In contrast, clustered attention places no such restriction.

[R1] Analysis of clusters and Set Transformers: Thank you for the suggestions, we will add further analysis of the clustering in the supplementary material as suggested. We will also add Set Transformers to the related work section.

[R1, R2] Experiments on other modalities. Is speech favorable to clustering?: Thank you for your comments. We would like to mention that our NLP approximation experiment for GLUE and SQuAD tasks in § 4.3 shows that clustered attention can capture arbitrarily complex attention patterns. In the future, we will also explore applications to NLP/vision tasks in the long context setting, as suggested.

[R1, R2] Other baselines to investigate: Longformer, Sparse Transformer, Pay-less-attention, and Local attention: We thank R1, R2 for the valuable suggestions. Sparse Transformer's reliance on the blocksparse library and the lack of documentation significantly hinder its evaluation on new tasks (evident by the lack of comparisons in the community). Longformer is a very recent work that we will compare with in the future. Upon R1's suggestion, we evaluate local attention on the synthetic copy task (§ 3.2 in the supplementary). We observe that for a sequence of length 128, local attention with context of 32 and 4 layers cannot solve the task whereas all clustered attention variants can solve it perfectly with 30 clusters.

[R3] Is sparse dot product used in improved-clustered attention practically efficient?: We refer R3 to Fig. 3 of the supplementary, where we present the time-memory benchmark. Notably, improved attention is faster than vanilla for sequence lengths greater than 1024. Unlike general sparse dot products, our implementation exploits the fact that all queries in a cluster use the same top-k keys. This allows us to cache the keys in CUDA shared memory for better performance. We will add details about our implementation in the supplementary material.

[R3] If C grows with N, then the method has quadratic complexity: We agree with R3 that C should be small for good performance. Note that, C is not necessarily a function of N as it depends heavily on the task. We would like to refer R3 to an ablation on the relationship between clusters and sequence length in § 3.2 of the supplementary. For the masked copy task, improved-clustered solves the task for every sequence length (up to 512) with just 15 clusters. Similarly, for SQuAD approximation, using only 25 clusters for a sequence length of 380 leads to good performance.

[R3] Training time (table 2) is only slightly faster with worse performance than the baseline?: While for CTC loss (table 2), the training time improvements were only small, for LF-MMI loss (table 3), the improvements were significant. Also note that while the average sequence lengths for WSJ and Switchboard are 780 and 530, the maximum lengths are 2500 and 3850 respectively. Finally, during inference under fixed computational budget, improved-clustered is consistently better than the baselines (Fig. 1).

[R3] Is clustering complexity O(NCD) so bad? What is actual speed improvement on RoBERTa? Minor comments: Thank you for the suggestion, we agree that for long sequences, k-means with euclidean distance O(NCD) could improve performance due to better clustering. For RoBERTa approximation, vanilla attention is faster due to small sequences. The purpose of the experiment is to understand the ability of the clustered attention variants to capture complex attention distributions using only a few clusters. We will make it clear in the camera-ready version that vanilla attention is faster in this experiment. We also want to thank R3 for pointing out the mistakes, we will fix them.