Experiment	Dataset	Model	Tokens	#	C	Perf.
Comp. with Longformer	Hyperpartisan news	Longformer	4096	NA	NA	94.8%
		SMYRF-RoBERTa	512	4	64	**97.2%**
Larger models	CoLA | QQP	SMYRF-BERT (**large**)	128	2	32	60.4% | 90.2%
Comp. with Reformer / LSH Ablation	IMDB (3 epochs)	SMYRF-RoBERTa	512	8	32	**93.7 %**
		E2LSH	512	8	32	89.3%
		Reformer (Cross-Polytope LSH)	512	8	32	88.7%

New Experiments: We thank the reviewers for their valuable comments! We see that reviewers appreciated the advantages, especially that we can use this method as a drop-in replacement in pre-trained NLP and CV models.

[R1], [R2], [R3], [R4]: in response to your requests for further comparisons, we first compare with the Longformer, which reports scores on downstream tasks. As shown in the table, we significantly outperform the Longformer on Hyperpartisan News, even with significantly smaller context length. [R2]: *"There is no ablation study for Novel Asymmetric Transformations. Unclear if the proposed method is better than only using LSH. [..] The authors need to have some fair comparison with Reformer or LSH"*. Thank you for the suggestions. As shown in the table, our novel ALSH significantly outperforms the E2LSH and the Reformer LSH scheme. [R3]: *"can the method be applied on longer sequences?"* Yes. In fact, we managed to train BigGAN from scratch with **16K tokens** (see Table 5) and **65k** tokens (see section 4.3 , 7.2 of supp.). Unfortunately, many pre-trained NLP models have been trained with maximum positional embeddings at 512 tokens, which prohibits finetuning in larger inputs. This is one of the main reasons we could not directly compare with Reformer / Routing Transformer. [R4]: *"It's unclear whether the proposed method performs worse along with the increased model size."* We run additional experiments on some GLUE tasks to show that our method works well even for bigger model sizes (see above Table). SMYRF-BERT large consistently outperforms SMYRF-BERT base (see also Table 2). We used SMYRF on all 24 attention layers vs 12 for base. [R4]: *"Most gains come from RTE which is a small dataset [..], on the most important task MNLI, the performance degrades significantly"*. The dependency between dataset size and performance is unclear. For example, QQP is a fairly big dataset, in which SMYRF outperforms vanilla BERT while using 50% less memory (see Table 2). We respectfully disagree that our method creates a significant performance degradation on MNLI: with 50% less memory our reported score is less than 1% shy of the performance of BERT. To further address this, we performed hyperparameter search for MNLI (lr=$3e-5$, batch=8, 5 epochs) and we obtained **85.02%** acc., which is *better* than the 84.43% acc. of BERT. We plan to include and expand the results of the above table in the Camera Ready version.

On Distillation: [R2], [R4] mentioned distillation as an alternative to reduce memory. Knowledge distillation creates smaller models while our method allows larger inputs. The two innovations are not mutually exclusive. If we still compare, DistilBERT reports *worse* scores in all GLUE tasks and reduces memory by only 40%. There are plenty other orthogonal methods to save memory, such as reversible layers. We plan to discuss them, as suggested by [R2].

Theorem 1: We would like to thank [R3] for detailed feedback. We will use the proposed name. [R3]: *"line 88 (supp. material) you forgot a minus sign."* Correct, that is a typo: the last three equations are **max**, (see Lemma 1, Eq. 3). [R3] *"What is q_D, $q_{C\,D}$ and q_C in lines 87-88 of the appendix?"* Indeed our notation needs explanation. q_D denotes the softmax denominator for query q, i.e. $\sum_{k\in\mathcal{K}} e^{q\cdot k}$. Similarly, q_C (which we mistyped as $q_{C\,D}$ before L88) denotes the softmax denominator of the cluster for query q, i.e. $\sum_{k\in C_t^L} e^{q\cdot k}$. We will update these. [R3]: *"It is not clear that the proof works for ϵ=0"*. Our argument does not require $\epsilon = 0$ or any infinities actually: There exist finite a, ϵ that suffice (they depend on the input vectors Q, K). We only need q_C to be sufficiently close to q_D so that the maximizers of the two problems are the same. For any given input instance this is a finite difference. We will rewrite our proof to avoid infinities and have a cleaner argument. Finally, [R3] points out connections to biclustering and co-clustering. We will discuss this and cite the relevant work, thank you for pointing this interesting connection to classical work.

Other [R3]: *"Does your architecture support causal masking?"* Yes. If a token gets clustered with tokens from the future we just zero these entries in the softmax. When a token is clustered only with tokens from the future, we only allow this token to attend to itself. [R3]: *"the paper is not clear about how L is chosen."* For $O(N \log N)$ complexity, L should be $O(N)$, (mentioned in L173). In practice, we try to minimize the number of queries per cluster. Choosing $O(1)$ queries per cluster, brings L to $O(N)$. We state the number of queries per cluster in almost all our experiments (see column C of Tables 1, 2, 3, 5) to show how different choices of L impact performance. We will follow the advice of [R3] and explicitly discuss this in the Complexity Analysis section. [R4]: *"It's more insightful to show that the proposed method can also work well in the pre-training setting."* We agree with [R4]. That is why we included in the paper pre-training results for BigGAN (see Table 5). Our budget did not allow additional pre-training experiments for NLP. [R4]: *"It's unclear how GPU friendly the method is."* Our solution is as GPU friendly as Reformer's attention, since the codebase only differs in the LSH scheme. Our method is more useful in terms of speed for large sequences (see Fig. 6 of our supp and Fig. 5 of Reformer). We will report GPU hours as requested by [R4]. [R4]: *"More analysis would help to understand the limitations."* Please refer to "Things that did not work section". Finally, all typos and figure suggestions will be addressed as suggested.