Thanks for the reviews! We have run several new experiments in response to reviewer comments. See detailed replies to questions below:

[R1] : impact of top-k on mode collapse compared to baselines. To show that Top-k sampling reduces mode-collapse in GANs, we compute the standard *Number of statistically Distinct Bins* (NDB/K) metric - a lower score on this metric is better. **On CIFAR-10, top-k sampling improves the NDB number from 0.75 to 0.60, using a SAGAN baseline. This suggests that top-k sampling actually reduces mode-collapse.** We'll add this discussion to the paper.

	1	2	3	4	5+
GAN	40.8	26.3	12.4	7.8	12.7
Top-k GAN	74.9	14.7	2.8	2.3	5.3
Target data	68.2	27.2	4.2	0.2	0.1

Table 1: Fraction of samples between 0 and 1 standard deviations away from their closest mode, and between 1 and 2, etc. Top-k sampling reproduces the underlying distribution much more faithfully.

[R2] : The novelty of the method is rather incremental The main contribution of this method is to present a previously unknown property of GAN training, which significantly hinders GAN training across different GAN architectures and objectives. We also study one reason for the existence of the phenomenon in the Mixture of Gaussian setting, where we show how the samples travel away from the nearest mode, when a gradient update is applied on the *worst* samples in a batch. We also show a significant boost in performance for many state-of-the-art GANs, namely BigGAN and ICRGAN, which further strengthens the importance of the proposed technique.

[R2] : No comparison of similar works using critics for rejection sampling

We performed new experiments to address this point: First, we perform Discriminator Rejection Sampling (DRS) on a vanilla SAGAN on CIFAR-10. This improves the FID from 19.1 to 18.2, as you'd expect. Then, we performed DRS on a SAGAN trained with top-k updates. This improves the FID from 17.8 to 17.2. **This shows that top-k training and post-processing techniques like DRS are not in competition - you can do both to further improve results!**

[R3] : There are two common methods, when the training is hard to converge, abandoning hard cases can make the training more stable, while when the task is easy to converge, paying more attention to hard cases to achieve better performance. Please refer to the NDB table above, which suggests that using top-k training does not result in GANs dropping hard-to-model data. The worst samples in a batch are harmful to training, since the Discriminator is unable to provide meaningful gradients, not because the data itself is hard-to-model.

[R3] : how different annealing strategies influence performance. Table 5 in the submission has this information.

[R4] : Would be nice if the authors can present more convincing results, for example, the comparison with other SOTA methods. We perform two experiments with models that are the state-of-the-art including BigGAN and ICR-GAN, and are able to get significant improvements on both models. As far as we know, **Our results from Table 4 are the current state-of-the-art for CIFAR-10.**

[R4] : When applied to large dataset like ImageNet, the improvement is not that significant. Since ImageNet is a very large dataset, there was no hyperparameter tuning done on ImageNet, and the exact same hyperparameters for CIFAR-10 were used for ImageNet. It's likely possible to improve the results even further given more tuning. Also, the improvement in FID from 19.98 to 18.44 is quite significant! Reducing FID from 100 to 99 is exponentially easier than reducing it from 20 to 19. Consider also that the addition is only one line of code.

[R4] : I would expect the authors to add an ablation study of the comparison using different batchsizes.

Table 2: The effect of batch-size and Top-k sampling. Top-k sampling's usefulness does not seem to drop off as batch size increases.

SAGAN (batch-size=64)	Top-k SAGAN (batch-size=64)	SAGAN (batch-size=128)	Top-k SAGAN (batch-size=128)	SAGAN (batch-size=256)	Top-k SAGAN (batch-size=256)
21.1	**19.8**	19.0	**17.9**	18.6	**17.4**