We thank the reviewers for their comments, which we found to be quite helpful. We have strengthened our submission with this feedback, and we believe the below addresses the main concerns identified in our submission.

First concern is the reviewer believed that we just borrow sparsification algorithms designed for image classification networks, and use them to train generative models. So he thought the baselines are not convincing.

We think this is a misunderstanding. For the experiments to compare with our method, we all follow the general settings and schedules when training the generative models, i.e., training in an adversarial fashion, as shown in ***Table 1***. Just as comments from other reviewers, we apply the common sparsification algorithms in these baselines is to explain why existing techniques work for models trained on classification tasks but not GANs. So all the baselines are convincing.

Second concern is the reviewer thought we are missing the clear evidence of some claims about the loss curves.

Actually, the detailed evidence of loss curves are already provided in ***Appendix***, ***Section A.1***. And the detailed explains are already provided in ***Discussion*** part in ***Section 3*** of the manuscript. For example, in which situations, loss curves that look identical to the baseline training will also lead to the bad compression, and the discriminator falls into a low-entropy solution that will cause mode collapse.

Third concern is the reviewer thought we overload the term “self-supervised”.

We thank the reviewers for pointing this out. We think the better term to describe our approach can be “self-tuning", “self-correcting", “autoregulative".

Fourth concern is the reviewer preferred more mathematically analysis on the performance boost in compression.

We provide the analysis from the ***Bayes theory*** perspective. The three deep neural networks in the ***GAN*** compression task are the original generative model G_O, the compressed generative model G_C, and the discriminative model D. Given x as the input of the generative networks, we can denote the generative outputs as $G_O(x)$ and $G_C(x)$.

We use x_i and x_j to represent two training samples from different categories. Our target is to push closer the generative outputs of the original and compressed generative models with the samples from the same categories, while to push apart the outputs of these two models with the samples from different categories. ***KL*** divergence is applied to measure the difference between two generative representations. Ideally, the target can be denoted with the following formulas.

$$\boldsymbol{KL}(G_O(x_i), G_C(x_i)) \to 0, \qquad \boldsymbol{KL}(G_O(x_j), G_C(x_i)) \to \infty \tag{1}$$

We define a latent variable S which represents whether the two input samples are from similar ($S = 1$) or different ($S = 0$) categories. For ease of notation, we define the event U to denote the generative representations between the G_O and G_C models are similar, and the event V denotes the D model regards the generative results are similar, i.e.,

$$\begin{aligned} U \Rightarrow G_O(x) \doteq G_C(x), &\qquad \overline{U} \Rightarrow G_O(x) \neq G_C(x) \\ V \Rightarrow D(G_O(x)) \doteq D(G_C(x)), &\qquad \overline{V} \Rightarrow D(G_O(x)) \neq D(G_C(x)) \end{aligned} \tag{2}$$

According to the total probability formula, for the whole GAN compression process:

$$\begin{aligned} P(S=1) = &P(S=1 \mid U, V)P(U,V) + P(S=1 \mid \overline{U}, V)P(\overline{U}, V) \\ &+ P(S=1 \mid U, \overline{V})P(U, \overline{V}) + P(S=1 \mid \overline{U}, \overline{V})P(\overline{U}, \overline{V}) \end{aligned} \tag{3}$$

If the discriminator is initialized by the well-trained model, then the probability of joint distribution for event U and V will be close to $P(U)$, while the probability of joint distribution for $\overline{U}, V$ and $U, \overline{V}$ will be close to 0, simplify ***(3)*** as:

$$P(S=1) = P(S=1 \mid U)P(U) + P(S=1 \mid \overline{U})P(\overline{U}) \tag{4}$$

Because the G_C model is initialized by G_O, so the second item in ***formula (4)*** has much less influence.

If the discriminator is randomly initialized as the original GAN baseline, then U and V can be regarded as the relative independent events. So the four items in ***formula (3)*** have a certain probability of occurrence.

Because the first item in ***formula (3)*** and ***(4)*** is our learning target. Our proposed method keeps the same total probability but changes the probability distribution. Because the optimization process during the learning cannot guarantee to find the global optimum. So an easier learning target has a higher expectation to be achieved during the same compression and optimization process. (We will extend to more rigorous prove without the one-page limitation.)

Improvements: We will add these minor improvements suggested by the reviewers in the final camera-ready version.

1. Provide the same level of thorough quantitative and qualitative results comparing to the baseline techniques for at least one more GAN architecture and dataset. (We will add them in ***Appendix***.)

2. Improve the captions which will be helpful for readers who like to skim figures before deciding to read a paper.

3. Add the experiment setting as training a small & dense network from scratch, but with the discriminator initialized as the trained discriminator.