We thank the reviewers for their comments and suggestions. We address their comments separately.

Reviewer #1:

1.1 It is unclear how the bias is handled and the relation to inexact gradient method.

Reply: First, in this paper we are handling the mean-squared error of estimators, which can be decomposed as $\mathbf{E}[\|\tilde{\nabla} F(x_i^t) - \nabla F(x_i^t)\|^2] = \|\nabla F(x_i^t) - \mathbf{E}[\tilde{\nabla} F(x_i^t)]\|^2 + \mathbf{E}[\|\tilde{\nabla} F(x_i^t) - \mathbf{E}[\tilde{\nabla} F(x_i^t)]\|^2]$. The first term is the squared norm of bias (nonzero in our case) and the second term is the variance. Therefore, our procedure is controlling both of them (see proof of Lemma 1 in Appendix). We shall emphasize this in revision. Second, our method can be considered as an inexact gradient method in the general sense. However, the additive inexactness are carefully controlled by the amount of descent to yield desirable complexity, which is closely related to the property of the estimator. Thus we cannot directly use the standard analysis of inexact gradient method, which will lead to worse complexity.

1.2 The step size of the algorithm needs to be small in typical non-convex optimization, usually depends on ϵ.

Reply: Even for non-convex optimization, constant step size can be used if the objective function is smooth. Small or diminishing step sizes are mostly required in stochastic optimization to combat noise in the stochastic gradients, for both convex and non-convex optimization. We are able to use constant step size in the stochastic optimization setting because of much stronger assumption: we assume each realization of $g_\xi(\cdot)$ is a smooth function, not merely its expectation $E_\xi[g_\xi(\cdot)]$ as in classical stochastic optimization. This assumption is key to effective variance reduction in stochastic optimization, and is satisfied in most machine learning problems. We will elaborate on it in the revision.

Reviewer #3:

3.1 Significance of improvement and reference to [34].

The problems we consider are the same class of problems addressed in [34], thus lead to similar motivation and introduction. We will revise the introduction to be more concise and rely on reference to [34]. While [34] first showed that variance reduction techniques (SAGA/SVRG) can be used to handle biased gradient estimators in composite stochastic optimization problems, this paper improves the dependence on n from $n^{2/3}$ to $n^{1/2}$ using a different estimator SARAH/SPIDER. From theoretical perspective, the improvement is significant for large n. More importantly, it reaches the lower bound for the considered problem class in the inner finite-sum case (see next point).

3.2 Discussion of lower bound in more depth than line 99.

Reference [8] showed that for nonconvex smooth finite-sum optimization problems (without the outer composition), the lower bound on sample complexity of the component gradients is $O(n^{1/2}\epsilon^{-1})$. That is a special case of the composite problem we consider, where the inner smooth mappings are functions (range dimension is 1) and the outer composition is the identity function. From this perspective, our result also reaches the lower bound for the more general nonlinear composite problems. We will clarify further in the revision, especially with the additional page allowed if accepted.

3.2 Questions on the experiments. We agree with the reviewer that more exhaustive search and tuning of parameters may be needed to compare the bests of different algorithms. However, the main purpose of our experiments are to demonstrate some basic behaviors of the algorithms on a few simple examples. Remember that the complexities cited for different algorithms are their theoretical upper bounds. It is not clear if SVRG or SAGA based estimators can achieve the same complexity as CIVR. At least in the convex finite-sum case, their complexities are the same. Nevertheless, we will do some additional experiments to gain more understanding.

Reviewer #4:

4.1 On preventing restart. First, our restart scheme in Section 4 does not use different ϵ at different periods (outer loop). We use the desired ϵ to set all algorithmic parameters once and do not change later. The only reason we use restart is that the output of Algorithm 1 is chosen randomly from all past iterates within one period, which we need to use to start the next period for analysis (see Proof of Theorem 5 in Appendix C.1.) If we can use the last iterate of each period, then there is no need to perform "restart". Nevertheless, it can be avoided by using pre-generated stopping times. Note that we can predetermine period length T and epoch length τ ($\tau_1 = \cdots = \tau_T$). The output is drawn uniformly from the $T\tau$ iterates. Therefore, we can first uniformly randomly generate the "stop time" $(\bar{i}, \bar{t})$. Then as soon as the algorithm arrives this time, we "restart" seamlessly. We shall add this remark to the revision.

4.2 The sensitivity on η. Our bounds on η in the theorems are to guarantee convergence and order of complexity. Smaller η can be used, but will encounter a performance penalty, which is explicit in Theorem 1, 2 and 3 on the bounds in (22), (23) and (24) respectively. For theorems in Section 4, small η will make $T \propto 1/\eta$ larger (also explicit in the theorems). This will translate into total iteration cost, which we shall clarify in the revision.

4.3 The "loopless" version (for inner loop, as the outerloop for restart is addressed in 4.1).

We thank the reviewer for pointing out this interesting literature. According to our analysis, this technique can indeed by applied, and we also get the same complexity. However, different from the convex problem in the "loopless" paper, in the non-convex case the "loopless" proof is a bit more complicated then the current analysis. We will point out this potentially interesting variant of the our method in the revised version.