We thank the reviewers for appreciating our work, and for their insightful and constructive comments.

Response to Reviewer #1:

1. Regarding the innovation of this paper.

A1: For the comparison with [15], Ge et al. [15] uses the SVRG estimator $v_t = \frac{1}{b}\sum_{i\in I_b}\big(\nabla f_i(x_t) - \nabla f_i(\tilde{x})\big) + \nabla f(\tilde{x})$ which reuses a fixed snapshot full gradient $\nabla f(\tilde{x})$, and achieves a suboptimal convergence $\tilde{O}(n^{2/3}/\epsilon^2)$. Here we use the stochastic recursive gradient estimator $v_t = \frac{1}{b}\sum_{i\in I_b}\big(\nabla f_i(x_t) - \nabla f_i(x_{t-1})\big) + v_{t-1}$ to obtain the improved optimal convergence $\tilde{O}(n^{1/2}/\epsilon^2)$ for escaping saddle points which matches the lower bound $\Omega(n^{1/2}/\epsilon^2)$. Moreover, we also have discussed with the authors of [15], we cannot directly add a perturbation step to SPIDER [11] for escaping saddle points. The original SPIDER [11] uses an additional negative-curvature search subroutine (Neon2 [4]) to escape saddle points. Thus, in this paper, we use a few simple but effective modifications to obtain a simple algorithm for achieving the new convergence results. There are three differences between our SSRGD algorithm and SPIDER. 1) SPIDER [11] uses the normalized gradient update $x_{t+1} = x_t - \eta(v_t/\|v_t\|)$, while our SSRGD uses the direct update $x_{t+1} = x_t - \eta v_t$, where v_t is the gradient direction estimator. 2) SPIDER uses a very small step size $\eta = O(\epsilon/L)$, while our SSRGD uses the more standard step size $\eta = O(1/L)$. 3) SPIDER uses an additional negative-curvature search subroutine (e.g., Neon2 [4]) to escape saddle points, while our SSRGD directly escapes saddle points by adding a uniform perturbation sometimes. These key differences are crucial for achieving the new convergence result $\tilde{O}(n^{1/2}/\epsilon^2)$ and also show that our SSRGD is more attractive in practice. Intuitively, in the large gradient situations (e.g., at the beginning phase), SPIDER goes very slowly due to the normalized gradient update. Also note that SPIDER can still go slowly even in the small gradient situations due to its tiny step size. As a result, SPIDER may get worse convergence results in some situations (see the following response A2). Moreover, our SSRGD which uses direct update (non-normalized), standard step size and simple perturbation step (no negative-curvature search subroutine) is more attractive and easy to implement in practice. Besides, we believe our simple/natural analysis will be useful for future work.

2. Regarding the convergence results.

A2: We would like to point out that our convergence rate is not worse than SPIDER [11]. Moreover, our SSRGD is better than SPIDER if $\delta \le 1/\sqrt{n}$ (i.e., high second-order accuracy situation) due to the $1/\delta^5$ term in SPIDER (see our Table 1). Concretely, for achieving an (ϵ, δ)-second-order stationary point ($\|\nabla f(x)\| \le \epsilon$ and $\lambda_{\min}(\nabla^2 f(x)) \ge -\delta$), the last three algorithms in our Table 1 can obtain the best rate $\tilde{O}(n^{1/2}/\epsilon^2)$ in a large range of ϵ and δ. SNVRG+Neon2 [32] and our SSRGD achieve the rate $\tilde{O}(n^{1/2}/\epsilon^2)$ if $1/\epsilon \ge 1/\delta^2$ and $n \le 1/\epsilon$. SPIDER+Neon2 [11] achieves the rate $\tilde{O}(n^{1/2}/\epsilon^2)$ if $1/\epsilon \ge 1/\delta^2$ and $n \ge 1/\epsilon$. Similarly, for the online case (Table 2), SNVRG+Neon2 [32], SPIDER+Neon2 [11] and our SSRGD can also obtain the best rate $\tilde{O}(1/\epsilon^3)$ in a large range of ϵ and δ for achieving an (ϵ, δ)-second-order stationary point, i.e., SNVRG+Neon2 [32] and our SSRGD achieve the best rate $\tilde{O}(1/\epsilon^3)$ if $1/\delta^3 \le 1/\epsilon$, and SPIDER+Neon2 [11] achieves the best rate $\tilde{O}(1/\epsilon^3)$ if $1/\delta^2 \le 1/\epsilon$.

Response to Reviewer #2:

Future directions: The achieved convergence rate $O(n^{1/2}/\epsilon^2)$ matches the lower bound $\Omega(n^{1/2}/\epsilon^2)$ in finite-sum case. However, whether $\Omega(1/\epsilon^3)$ is the lower bound for online case is unknown. This should be an important future problem. The combination of the stabilized trick proposed in Ge et al. [15] is also indeed an interesting extension.

Experiments: We will try to add some experiments although previous related papers also did not do the experiments.

Response to Reviewer #3:

Regarding the difficulties to avoid using the negative-curvature search subroutine: Previous algorithms use the negative-curvature search subroutine (e.g. Neon2) to find the approximate smallest eigenvector of the Hessian (i.e., decreasing direction) near saddle points. However, if we just use a uniform perturbation step, it is not very easy to obtain the smallest eigenvector direction since we only roughly have $O(r/\sqrt{d})$ amount in the smallest eigenvector direction in a uniform perturbation ball with radius r in $\mathbb{R}^d$ space. In our analysis, by simultaneously bounding the coupled distance and coupled variance, we can show that the amount in the smallest eigenvector direction will increase exponentially and thus avoid the negative-curvature search subroutine.

Response to Reviewer #4:

Subroutine: The subroutine Neon [31] extracts the negative curvature based on a perturbation step, power method and Nesterov's accelerated gradient method. Neon2 [4] is based on a perturbation step, Oja's algorithm and Chebyshev polynomial. Thus they are at least more complicated in implementation than just a perturbation step as our SSRGD used. Please see A2 in Response to Reviewer #1 for the convergence comparison among the last three algorithms.

Optimal among simple algorithms: Note that Du et al. [10] showed that it is necessary to add the perturbation step for gradient descent to escape saddle points efficiently. Our SSRGD indeed just adds a perturbation step on the simple recursive gradient descent to achieve the optimal convergence $\tilde{O}(n^{1/2}/\epsilon^2)$. **Improvement/Comparison:** Intuitively, it can improve upon SVRG [21] since it uses more aggressive recursive gradient estimator instead of the conservative SVRG estimator which uses a fixed full gradient snapshot. Please see A1 in Response to Reviewer #1 for more details and a comparison with SPIDER [11], and see Response to Reviewer #3 for the discussion of saddle point iterations.