We thank the reviewers for appreciating our work, and for their constructive suggestions to improve its quality.

Response to Review #1: **Intuition on linear rate:** Varag achieves linear convergence rate when $m \geq D_0/\epsilon$, and sublinear rate when $m < D_0/\epsilon$, which relies on our selection of the inner loop size T_s. In our convergence analysis, we notice that the convergence rate is roughly in the order of $1/T_s$ (see Lemma 7), hence, if T_s increases exponentially, we can achieve linear convergence rate. Intuitively, it is reasonable to always increase T_s in order to avoid the full gradient computation when m is very large, i.e., $m \geq D_0/\epsilon$. It then stops increasing T_s when $T_s = m$, since the cost of full gradient computation is comparable to that required by m inner loops. We will add such discussions in the text. **Sampling method in experiments:** We use uniform sampling strategy to select f_i in all experiments. Indeed, theoretically the sampling distribution can be non-uniform, i.e., $q_i = L_i / \sum_{i=1}^m L_i$, which results in the optimal constant $L = \frac{1}{m}\sum_{i=1}^m L_i$ appearing in the convergence results. A uniform sampling, e.g., $q_i = \max L_i$, will lead to a constant factor slightly larger than L. Note that L_i can be estimated by performing maximum singular value decomposition of the Hessian. This is computationally efficient because only a rough estimation suffices. We appreciate the reviewers' comments and will add corresponding experiments and discussions in the experiment section.

Response to Review #2: **Introduction section:** We will add more examples and discussions on cases of strong convexity of f and stochastic finite-sum problems. $\mathbf{D_0}$ **in Table 2:** We will fix the footnote for D_0. **Sampling distribution:** See response to review #1 about a similar question. **Relation with other accelerated methods:** We pointed out in the footnote 1 that Catalyst requires restarting to achieve the optimal convergence rates, Katyusha needs to add perturbations to achieve optimal rates for smooth problems. We will expand these discussions and put them into main text. Note that we also compare Varag with Katyusha in details after we present Varag in Alg. 1.

Response to Review #3: 1. Thanks for pointing out this typo. x should be x^* in Equation (2.6).
2. We say that $O(m \log 1/\epsilon)$ is linear but not sublinear w.r.t. ϵ. We cannot replace m by D_0/ϵ because it leads to a too optimistic bound. Moreover, m is a constant independent of ϵ, and roughly every m gradient computations will increase 1 digit of accuracy, so we call it a linear rate. Indeed we admit that an $O((D_0/\epsilon)\log 1/\epsilon)$ bound would be better than $O(m \log 1/\epsilon)$ if $m > D_0/\epsilon$. We will add such discussions in respective places in the main text.

3. Thanks for this suggestion. Indeed one can assume each individual f_i is associated with a minibatch instead of a single piece of data. For the more general minibatch version, one can replace $G_t = (\nabla f_{i_t}(\underline{x}_t) - \nabla f_{i_t}(\tilde{x}))/(q_{i_t} m) + \tilde{g}$ (Line 7 of Algorithm 1) by $G_t = \frac{1}{b}\sum_{i_t \in S_b}(\nabla f_{i_t}(\underline{x}_t) - \nabla f_{i_t}(\tilde{x}))/(q_{i_t} m) + \tilde{g}$ with $|S_b| = b$ and adjust the appropriate parameters to obtain the minibatch Varag. We expect that the minibatch Varag will obtain the parallel linear speedup of factor b if minibatch size $b \leq \sqrt{m}$. We will incorporate such analysis into the revision of the paper.

4. We will update it as "Varag is the first accelerated randomized incremental gradient method that benefits from the strong convexity of the data-fidelity term to achieve the optimal linear convergence" to be more accurate.

5. Our Varag method is not adaptive and we will mention this explicitly in the later version. Note that the adaptivity of hyperparameters, i.e., smooth parameter L and strongly convex parameter μ (Varag only needs these two hyperparameters), is not the focus of our current Varag method. Varag uses a unified step-size policy to unify the convex problems with or without strong convexity, and directly achieve the best convergence rate for non-strongly convex problems. The adaptivity of hyperparameters is a good property for an algorithm, and we leave this as an interesting future extension of our work. We appreciate the reviewer's question and will add clarification into the revision.

6. Thanks for pointing out the recently Loopless SVRG paper [KHR2019] which removes the outer loop of SVRG by computing the full gradient with a small probability in each iteration. In the well conditioned/ big data case, our Varag switches to non-accelerated regime and achieves a linear convergence rate. We would like to point out that in non-accelerated regime, Varag, similar to Loopless SVRG, only needs to know L and does not require the knowledge of μ to set its parameters. Thus we believe that Varag will still work well in this case. We will some discussions about Varag's properties in this regime, as well as Loopless SVRG.

7. After briefly reading Loopless SVRG [KHR2019], we feel that Varag can also be possibly generalized into a loopless version similar to Loopless SVRG. We will discuss possible extensions of this method in the revised version.

8. We provide the theoretic suggestion of b_s and B_s by minimizing the stochastic gradient complexity: $\sum_s mB_s + \sum_s T_s b_s$. One can use other values for b_s and B_s and Varag can still converge to a stochastic ϵ-solution, but it may lead to a worse stochastic gradient complexity than our theoretic guarantee. We will add such discussions into the text.

9. Thanks for the constructive suggestions. We will try to add more experiments. Regarding the parameters, we only need two hyperparameters (i.e., the smooth parameter L and strongly convex parameter μ) to set all parameters in our experiment. We first use singular value decomposition (SVD) for the Hessian to compute L and μ at the beginning for all algorithms (this step is not included in the performance comparison). Then we use them to run all algorithms and compare their performance w.r.t. gradient computation. We will specify more details of the experiments and parameters setting in the revised version.

10. All typos will be addressed in the revised paper.