We appreciate the reviewers for the time and expertise they have invested in writing these constructive comments.

Reviewer #1

Q: *The lack of error bars. How does the method react to random initializations? Why aren't uncertainty shown?*

A: Thank you for your constructive suggestion, according to which we draw the error bars (mean $\pm$ std) to show how the method reacts to random initializations. Please see Panel (a) of Figure I for an example. We will use error bars to present our experimental results in the camera-ready version.

Q: *To increase my score even higher I need to be convinced that the theoretical result is a very substantial advance.*

A: Thanks. The significance of our theoretical contribution is to find a simple strategy to identify a single iterate from the iterate sequence with optimal convergence rates. While the existence of such an iterate is guaranteed by the fact that time-averaging gets optimal convergence, searching for such an iterate is non-trivial. Our method also has a potential to be applicable to other stochastic algorithms, e.g., stochastic dual averaging.

Reviewer #2

Q: *Algorithm 1 (Alg. 1): when I understand correctly, one has to calculate all iterates up to $t = 2T - 1$ and needs to store all iterates from $t = T$ up to $t = 2T - 1$.*

A: Thanks for the careful observation. Our description of Alg. 1 leaves an impression that it needs to store all iterates from $t = T$ up to $t = 2T - 1$ since we set T^* in line 17 of Alg. 1. However, this storage is indeed not required if we set $\mathbf{w}_{T^*} \leftarrow \mathbf{w}_t$ in line 17 of Alg. 1 (we only need $\mathbf{w}_{T^*}$ in practical implementation). We will address this in the revision.

Q: *Is the map $t \mapsto A_t$ monotone under (strong) convexity assumption? this refers to the choice of T^* in Algorithm 1*

A: Thanks for the query. Motivated by your comment, we run an experiment on SVM problems with a strongly convex objective to check the monotonicity of A_t. In Panel (b) of Figure I, we plot A_t as a function of t, from which we see that A_t is not a monotone function of t. We will mention it in the camera-ready version.

Q: *Wouldn't any $t \geq T^*$ also do the job? ... the best choice would possibly be the* $\arg\min$ *of all t satisfying the condition in l.16 (which is possibly the last iterate)*

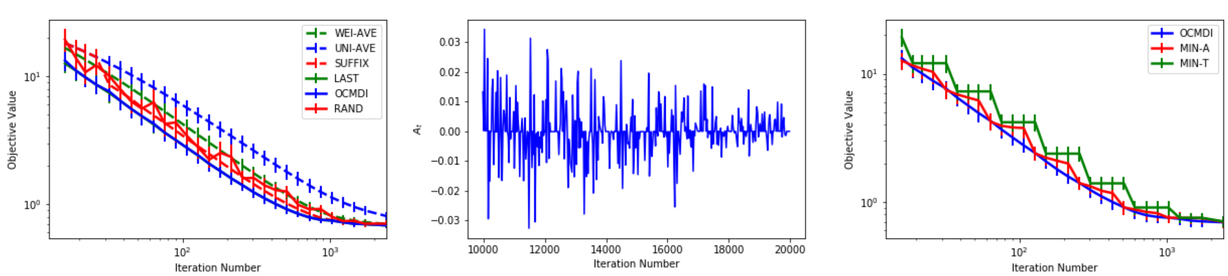
A: Thanks for the query. We conjecture that not all $t \geq T^*$ can achieve optimal convergence. The underlying reason is that $t \geq T^*$ may not necessarily satisfy the condition in line 16 of Alg. 1, which is required to get optimal convergence in our analysis.

Among all t satisfying the condition in line 16 of Alg. 1, the minimal t (MIN-T) has an appealing property of requiring the minimal computational cost, whose performance may be further improved if we update the model once encountering an $\mathbf{w}_{t'}$ satisfying the condition in line 16 of Alg. 1 with $t' > t$. The intuition is that the added computational cost may generally come along with a better model. This is the strategy adopted by SCMDI/OCMDI. Another strategy is to set T^* as the index whose associated $\triangle$ is minimal (MIN-A). The intuition is that the quality of $\mathbf{w}_t$ depends on $\triangle$ (please see line 243 of the paper). We run an experiment to show how OCMDI behaves versus MIN-T and MIN-A, and report results in Panel (c) of Figure I. We will add a comment in the camera-ready version.

Q: *Benefit compared to just taking averaging seems clear; I do not see the benefit compared to taking the last iterate*

A: Thank you for the comment. The benefit compared to taking the last iterate mainly consists in the theoretical property. Taking the last iterate can only achieve a suboptimal convergence rate with high probabilities (up to a $\log T$ factor), while our strategy can achieve the optimal convergence rate.

Reviewer #3: Thank you for your very positive comments.



(a) Error Bar Presentation (b) Monotonicity (c) Comparison on Selecting Iterates

Figure I: Experimental results of SPGD applied to SVM problems with the data Splice.