We would like to thank the four reviewers for their feedback.

Upon acceptance, we will include in the final version (a) *improved notations* and (b) *an improved presentation of related work*. We first discuss a few common concerns shared by **reviewer 1**, **reviewer 2**, **reviewer 3** and **reviewer 4**.

•• **Non convex bound:**: As pointed out by several reviewers, the non convex bound does not clearly show a dependence on the gradient prediction process. While being clear that in the convex case, predicting well the next gradient theoretically improves the bound (see Eq $(2)$), the non convex bound at least set a comparable state-of-the-art rate (as adam-type methods) in $\mathcal{O}(\sqrt{1/T})$. We acknowledge the need for a more precise consideration of the constants of both our method and AMSGrad to highlight not only an empirical edge of the optimistic update but also a theoretical one.

**Reviewer 1:** We thank the reviewer for valuable comments. Our point-to-point response is as follows:

**Convex regret bound:** For analysis purposes we presented the algorithm without projection step by assuming the compact assumption **H1**. Of course, this assumption needs to be verified and we partially did it for a model of interest that is a deep neural network, see Section 4.3. Adding projection steps is a neat idea to avoid having those issues but is not common in non convex optimization analyses, see references [5, 9, 14, 38].

**Numerical example:** We thank the reviewer for their remark on the numerical runs. The main motivation behind those plots is to show that adding an optimistic update to the vanilla AMSGrad actually speed up the convergence in terms of both losses and accuracies. Given the well-known advantages of Adam-type methods as ADAM or AMSGrad, we did not compare to slower methods "that does not have any of the extra features of AMSGrad" as written by the reviewer.

**Reviewer 2:** We thank the reviewer for valuable comments. A proofreading of the paper is being done as suggested and we give the following clarification:

**Wall clock times comparison:** We agree with the reviewer with the heavy computation that our gradient prediction process can represent. In the shown runs, and as precised Section 5.3, only $r = 5$ gradients are being used for the extrapolation step. Both memory and wall clock time are lightly impacted. To complete. Can we have the wall clock times plots?

**Reviewer 3:** We thank the reviewer for the thorough analysis. Our remarks are listed below:

**Gradient prediction algorithm:** We agree with the reviewer that a study of how well the gradient is predicted using the current method would be impactful. The scope of our paper being the stochastic optimization method itself, we invoked a simple but effective gradient prediction algorithm on the basis of reference [31] which shows great theoretical and empirical acceleration using such extrapolation. Of course, there is room for improvement regarding that prediction process and can be the object of further research papers.

To check if [31] gives theoretical guarantees on how well the prediction is

**Numerical evaluation:** It has been rightly noted that in Figure 3, the curves are still rising and thus convergence is not attained yet. Though, for illustrative purposes, the main idea is to show how faster our method is in the first epochs. The purpose of this method is not to achieve better generalization (*i.e.* reach better accuracies at convergence) but rather to show how less epochs are needed to achieve similar results as baselines. The learning rates have been tuned over a grid search and the best results have been reported. The choice of a constant learning rate was made to stick to our theoretical results. Runs with exponential decay or step decay can also be done for completeness.

**Reviewer 4:** We thank the reviewer for valuable comments and typos. Our response is as follows:

**Numerical experiments:** We only reported the average of the 5 runs but as the reviewer suggested we will report error bars in the rebuttal version of the paper.

We agree with the fact that our method empirically shows on Figure 2 and 3 better training performances (both in terms of loss and accuracy) but we must note how comparable and most of the time better than the baselines our method behaves at testing phase.