We sincerely thank the four reviewers for their valuable feedback. 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 do not have any of the extra features of AMSGrad" as noted 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:
Novelty of the contribution: Although combining gradient prediction to AMSGrad update seems natural, as pointed out in the first paragraph of Section 3, the algorithm does not reduce to AMSGrad when the prediction is null (contrary to Optimistic-FTRL). Rather, a focus has been made on how to integrate the prediction process which led to the two-stage algorithm OPT-AMSGrad where, first an auxiliary variable is updated and then the global model. This integration of the gradient prediction is represented Figure 1 (that will be improved as suggested by **Reviewer 4:**).
Wall clock times: There are several works considering applying Alg. 3 in deep learning, e.g. Nonlinear Acceleration of Deep Neural Networks, Scieur et al., with positive results. As noted in their paper, in practice extrapolation on CPU is faster than a forward pass on mini-batch and can be further accelerated on GPU. Moreover, note that in each iteration, we only change one past gradient, so we do not need to compute the whole linear system every time (but only a small portion). Therefore, it could be fairly efficient in practice. Secondly, the main focus of our paper is essentially the framework of integrating optimistic learning with AMSGrad. We choose Alg. 3 mainly because of the empirical success reported in prior works. The choice of gradient prediction method is actually flexible. So, OPT-AMSGrad will definitely benefit from an algorithm with faster running time and good prediction quality. This is more related to acceleration literature and we regard this as an interesting line of future research.

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. As we replied to Reviewer 2, we choose Alg. 3 mainly because that its success in deep nets has been observed in some prior works. Of course, there is room for improvement regarding this prediction process and can be the object of further research papers.
Numerical evaluation: For MNIST-image the test accuracy has been flat, and for the other two it is almost flat though slightly increasing. Yet, 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 accuracy 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 focused on running different methods with same initialization for a fair comparison, and OPT-AMSGrad consistently outperforms vanilla AMSGrad with same initialization. As the reviewer kindly suggested, we will report error bars in the rebuttal version of the paper to show the advantage of OPT-AMSGrad with random runs in practice. 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. • Typos and unclear figures will be changed according to the additional feedbacks. • Figure 1 ought to represent the two-stage method and will be simplified in a linear manner as suggested. • AMSGrad and OPT-AMSGrad can indeed be written jointly (to highlight our contribution we decided to separate them). • The termination number T is classical to derive theoretical results in nonconvex optimization (see [Ghadimi & Lan, 2013]). • The bracket notation in H3 is the inner product notation, this will be specified.