We thank the reviewers for taking the time to read and comment on our submission despite these challenging times.

Reviewer #1. You wrote: "This work studies an important problem ... The paper is mostly clearly written and most of the proofs in the paper seem correct." Thanks for acknowledging the strengths of our submission.

- *Regarding major technical mistake*. There is no mistake here. We are simply using the fact that the zero-one loss is bounded from above by the logistic loss, and taking expectations on both sides. This is what gives the inequality after line 259. What it states is that the expected zero-one loss, i.e., the probability of misclassification, is bounded by expected logistic loss; there is no *training error* involved in this expression. A detailed derivation is in the Appendix (look for Proof of Theorem 4.3 on lines 559-561).
- *No generalization analysis*. Theorems 4.3 and 4.6 are our generalization error bounds. Instead of analyzing the Rademacher complexity, we use a martingale concentration inequality to bound the generalization error. Please see the Proof of Theorem 4.6. in the appendix for more details.
- Our generalization guarantees do not *worsen* with the gradient updates, they *improve*. In both Theorem 4.3 as well as Theorem 4.6, we bound the generalization error by $\tilde{\mathcal{O}}(1/T)$, which implies that the generalization error decreases with T, i.e., with the number of gradient updates in Algorithm 1.
- The constant strategy $\mathrm{W}_t = \mathrm{W}_1$ would incur a constant error at each iteration, and thus not decay with T. In sharp contrast, the error of the iterates of Algorithm 1 decays as $\tilde{\mathcal{O}}(1/T)$. The advantage of Algorithm 1 over the constant strategy shows up when we apply Lemma 5.4 to the right hand side of the inequality given by Lemma 5.3.
- Thanks for your comment on Nagarajan and Kolter – we will make the *uniform convergence* discussion explicit in the final version of our paper to better convey the following quote from their paper: "... uniform convergence provably cannot "explain generalization" – even if we take into account the implicit bias of GD to the fullest extent possible."
- Regarding your minor comments, 1) we will extend the language in Remark 4.5 to emphasize on computational learning theoretic advantages of dropout over SGD; 2) we will make it more clear and precise; 3-4) thanks for catching the typos, p should be q.

We hope that our response has clarified all of your technical concerns, in which case, please reconsider your score.

Reviewer #2. Thanks for acknowledging the novelty of our results.

- The definition of logistic loss on line 123 is not wrong; feel free to refer to any ML book. You can also refer to the prior work, e.g., [Cao and Gu, 2019], [Ji and Telgarsky, 2019], etc. It is typical to define a loss function as $\ell(y, f(x))$ and overload the notation to write $\ell(z) = \ell(yf(x))$; e.g., for logistic loss this would be $\log(1 + \exp(-yf(x)))$. Please see the definition of hinge loss for another example.
- As we state in Remark 4.4, with high probability, dropout iterates will not violate the max-norm constraints, i.e., these constraints are not active in a typical run of dropout whatsoever. We only need the max-norm constraints to give generalization error bounds in expectation.
- Yes, $k \in \{1, \ldots, T\}$ represents an iterate. Linearization means we approximate a function with its best linear approximation, given in terms of the gradient of the function at that point. This is what is happening on lines 167-169. Neural tangent kernel (NTK) analysis is based on using ∇g_t as features; see the preliminaries on lines 141 – 151.
- Proof of basic fact on line 174:

$$\langle \nabla g_t(\mathrm{W}_t), \mathrm{W}_t \rangle = \sum_{r=1}^{m} \langle \frac{1}{\sqrt{m}} a_r b_{r,t} \mathbb{I}\{\mathrm{w}_{r,t}^\top \mathrm{x}_t \geq 0\} \mathrm{x}_t, \mathrm{w}_{r,t} \rangle = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r b_{r,t} \sigma(\mathrm{w}_{r,t}^\top \mathrm{x}_t) = \frac{1}{\sqrt{m}} \mathrm{a}^\top \mathrm{B}_t \sigma(\mathrm{W}_t \mathrm{x}_t) = g_t(\mathrm{W}_t)$$

- The weights returned by the algorithm are scaled by q to account for the dropout. Hence, the risk takes the argument $q\mathrm{W}_t$. We make the following remark on lines 139 – 140: "at test time, the weights are multiplied by q so as to make sure that the output at test time is on par with the expected output at training time."

We hope that our response has clarified all of your technical concerns, in which case, please reconsider your score.

Reviewer #3. Thanks for your encouraging comments! We define the *neural tangent feature* $\phi_{\mathrm{x}} : \mathrm{z} \mapsto \mathrm{x}\mathbb{I}\{\mathrm{z}^\top \mathrm{x} > 0\}$ on line 150 after Eq. 1. Assumption 1 is simply a margin assumption in the RKHS of the NTK. It is a mild and reasonable assumption that the data becomes linearly separable after mapping it into a high-dimensional non-linear feature space; this idea has been the cornerstone of kernel methods using the RBF kernel, for example, and for learning with neural networks. Further, this assumption yields bounds under mild over-parametrization compared to other works.

We hope that our response has clarified all of your technical concerns, in which case, please reconsider your score.

Reviewer #4 Thanks for your positive review!