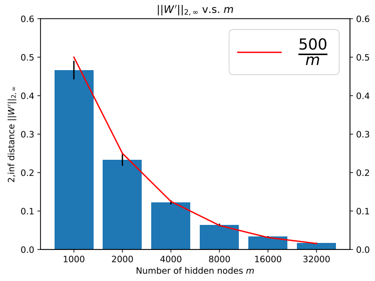
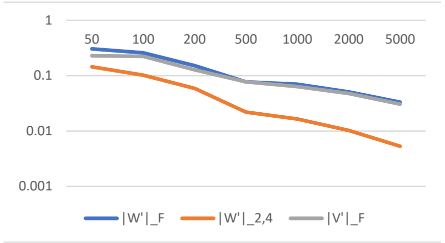
We thank all the reviewers for the time reading our paper! We will fix all the minor issues, and below we only address the main concerns. We restate those questions below.

- **R2:** The two-layer part of this submission might be similar to Daniely's?
  No. Daniely's result trains only the *last layer* of (any depth) network, and the changes of hidden layers are negligible (can be set zero). Daniely proves generalization using conjugate kernel, but it's unclear what *explicit functions* can be learned and what's the *explicit sample complexity*. In contrast, we consider target functions consisting of 2-layer networks, and show how they can be PAC-learned and what's the sample complexity.
- **R2:** Why the 3-layer result is *not based on NTK* (neural tangent kernel)?
  Recall $W = W_0 + W'$ and $V = V_0 + V'$ where W_0, V_0 are initializations. In NTK, by ignoring higher-order terms, the network is a linear function in (W', V'), so entries of W' will *never* be multiplied with V'. In contrast, see Lemma 6.10, we track $DV'DW'$ so V' and W' are multiplied together. This is non-linear so is not NTK. In fact, as we explained in the paper, our concept class (learnable by three-layer networks) is not captured by the NTK of a three layer network. We shall make it more clear in the revision. (We thank **R5** for carefully reading our paper and acknowledge "unlike prior work" we have considered "non-convex interactions between weight matrices.")
- **R2:** Does the sample complexity really depend polylog in the network size?
  Sorry for the confusion. What we mean is sample complexity *grows* polylog in m. Indeed, the sample complexity shares some poly terms with M_0, but as $m \geq M_0$, sample complexity only *grows* polylog in m.
- **R2+R4:** Why can't standard SGD (without noise) work?
  There are many reasons (see footnotes on Page 7). For instance, *all* existing escape-saddle point papers need noise. Since we rely on such existing work, we need noise for theoretical purpose.
- **R4:** Why increasing m supports more target functions? What is R?
  Sorry for the confusion and it is actually simpler than you may have thought. For any target function, there is a corresponding M_0 such that our theorem applies whenever $m \geq M_0$. Here, M_0 depends on the complexity notion introduced on line 122. In other words, if we increase M_0, there will be more functions to be supported by this threshold M_0. Finally, R only plays some role in our 3-layer theorem, where it allows the complexity to be composed with another complexity function.
- **R4:** Modify references to e.g. Table 1 in line 239. Great suggestion and we will do that!
- **R5** raises concerns about the significance of the generalization part of this paper.
  Although the generalization lemmas *on their own* are simple, they are not our main contribution (and constitute 5% of this paper). **Instead**, our main contribution is to make the convergence theorems and generalization lemmas *compatible*: SGD can find solutions with small norms so that generalization lemmas apply (almost independent of m). This cannot be done by combining any "convergence theorem" and any "generalization lemma". For instance, the prior work Allen-Zhu et al [2] proves convergence, but it is not compatible with our generalization lemmas.
- **R5** has concerns about the practical relevance of our generalization lemmas.
  - **R5**: What will happen if learning rate is independent of the parameter count?
    Under our (wlog.) choice of initialization $W, V \sim \mathcal{N}(0, 1/m)$ and output layer $\mathcal{N}(0, 1)$, it is ***not* a good idea** to use constant learning rate. For instance, in our 3-layer experiment below, any learning rate $lr \in [0.01, 0.1]$ works for $m = 50$, but any learning rate $lr \in (0.01, \infty)$ gives NaN error for $m \geq 5000$. In general, learning rate *depends on initialization*: if hidden weights are scaled up and output layer is scaled down, then the learning rate will increase.
  - **R5:** If learning rate decreases as m, does $\|W'\|$ and $\|V'\|$ decrease experimentally?
    Yes. For 2-layer network on MNIST with $lr = 400/m$ and target accuracy 95%, norm $\|W'\|_F / \|W_0\|_F$ decreases, see Li-Liang [30, Fig 5 of page 26 of NeurIPS camera ready]. Below in left figure, we show $\|W'\|_{2,\infty}$ also decreases. As another example with 3-layer networks, say inputs are random $\|x\|_2 = 1$ and the true label $y = x_1 \cdot x_2 + x_3 \cdot x_4$. Using our standard initialization, then $\|W'\|_F, \|W'\|_{2,4}, \|V'\|_F$ all decrease as m increases (for $lr = 1/m$, test errors drop below 0.003 within 30 epochs), see right figure. We have plugged them into the generalization formula and they do give very meaningful bounds on sample complexity even for large m.


||W'||_{2,∞} v.s. m
500/m
2,inf distance ||W'||_{2,∞}
Number of hidden nodes m
0.0
0.1
0.2
0.3
0.4
0.5
0.6
1000
2000
4000
8000
16000
32000


1
0.1
0.01
0.001
50
100
200
500
1000
2000
5000
|W'|_F
|W'|_2,4
|V'|_F