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 quickly point out in our revision we have already extended the lower bound to* all kernel methods*, not just correlation kernels. We plan to include that if the paper gets in.*

- **R2** thinks we might have left out some relevant works on this topic, by citing [37] as well as some learning theory papers on ReLU: namely, "[L]earning Neural Networks with Two Nonlinear Layers" and "[E]fficient Learning of Generalized Linear and Single Index Models with Isotonic Regression".

  We're afraid R2 has misunderstood our work. Our main contribution is to show "can **Neural Networks** learn some concept class in distribution-free, efficiently-learnable setting while kernel methods can not."

  - ReLU functions, in particular, are **not** known to be distribution-freely (efficient) learnable by **standard ReLU neural network**. In the cited prior works [L] and [E], the main learning algorithms are both isotonic regression, which is **not training a neural network** via SGD. Generally, our work is not about "there exist learning methods better than kernels", which is trivially true (in fact, linear regression using Gaussian elimination is not doable by kernel methods). Our main contribution is to show that "Neural Networks can be better learners than kernels (esp. NTK)". This was not known at all in the distribution-free setting, which we have emphasized in the introduction.
  - For [37], its separation works *only when* the solution (i.e., the network weights) corresponds to a "max-margin solution" (which means minimal-norm solution in our language). This is *not* necessarily efficiently learnable. We have made this point clear on lines 50-60 and given a counter example in Fig 1.

- **R2:** give more explanation for lower-bound (LB) construction, and can it be extended to output dimension 1?

  Due to space limitation, we didn't explain much about LB. Sorry. Our proof overview is short and on page 12, line 371-379. Intuitively, we constructed a "high-complexity polynomial" $G(F(x))$ that is degree k over d Boolean inputs. By Boolean analysis, we write the solution of kernel method in the Boolean Fourier basis, and then argue about its certain coefficients. (If we can learn $G(F(x))$ then some coefficient must be large, but it is too large so that the kernel method will memorize training data unless $d^{k/2}$ samples are given).

  Our lower bound can indeed be extended to networks with one dimension output. We choose $k \geq 2$ to present the simplest proof.

- **R4:** can result be extended to networks without skip connection? Our experiment suggests that non-ResNet cannot learn this concept class with comparable sample complexity, so having skip connection is helpful in our setting.

- **R4:** What's the practical relevance? It says "neural networks are better than kernels", but that's known in practice. While known in practice, we provide a *formal prove*. As we emphasized in the intro, perhaps to many practitioner's surprise, there was no supporting argument (in theory) before this work for "neural networks are better learners than kernels" at least in distribution-free (efficiently computable) setting. This is the setting most relevant to practice. This is also why our theoretical contribution is significant.

- **R4** has given suggestions regarding reformatting this paper. Thank you and we will keep this in mind.

- **R5** has concern about α^4 and worries that the risk (i.e. error) cannot go below α^4

  Our setting is different from traditional statistical learning, where the risk can go to 0 for a fixed concept class. Instead, for every $risk > 0$, we construct a different concept class. The correct way to interpret our paper is that

  "for every $\alpha > 0$ (which defines H and a concept class), NN can learn up to accuracy α^4, but NTK cannot learn up to accuracy α^2, if the same sample size (around $poly(1/\alpha)$) is given to both methods."

  Furthermore, Theorem 1 does give a meaningful bound when α is very small, since the constants in $O(\cdot)$ in Theorem 1 is *universal* and does not depend on α. Hence, ResNet can learn the target class with non-trivial accuracy with relatively few samples (think of a function with output around 1, then learning the function up to error 0.1 is very meaningful, although not asymptotic). More importantly, with these many samples, the error is *significantly smaller* than the best solution obtained by kernel methods. Our setting is not asymptotic, but still commonly considered in machine learning.

  Finally, we admit that supporting risk$\to 0$ (for a fixed concept class) would be an important future direction.

- **R5:** what's the overlap with Allen-Zhu et al. 2018?

  Short answer: lower bound has no overlap; for upper bound, Allen-Zhu et al [2][3] are essentially kernel methods; since we have proven lower bound for kernels, [2][3] must fail to produce α^4 error as we do in Theorem 1.

  Long answer: [2][3] focus on the setting where (more or less) the sign patterns of the ReLU's do not change during training. Hence, the network training process is similar to the NTK kernel regime (see line 35-42, as well as criticism in arXiv 1904.00687). In this paper, we show that ResNet can learn functions that are not learnable by NTK. This is due to the fact that the sign patterns change a lot, which requires new techniques. (Of course, to make this paper short, in proving upper bound we have adopted some known lemmas from [2].)