We appreciate all the reviewers' valuable comments. Here is our response to the major questions raised by the reviewers.

Reviewer #1:

Q: Regarding the notation $B_{w,\star}, B_w, \lambda_{\min}$, and $q(z)$. Missing $\sqrt{\log(Dm)}$ factor in Theorem 1.

A: We used $B_{w,\star}$ to highlight it's the radius of the ball that contains $\mathbf{W}_\star$, and B_w to denote an arbitrary radius (later we apply this result with B_w larger than $B_{w,\star}$). $\lambda_{\min}$ denotes the minimum eigenvalue. $q(z)$ in Line 174 is the dummy variable (under the $\min$) for a degree-k polynomial. ϵ in Algorithm 1 determines the choice of D (as in Theorem 2). We appreciate these suggestions and will make a supplementary table for all the symbols in our next version. Our Theorem 1 does contain a $\sqrt{\log(Dm)}$ factor (complete version is provided in Appendix Line 464 with $\sqrt{\log(Dm)}$ in the first term). We used $\widetilde{O}(\cdot)$ to hide dependency on any log factor in the theorem statement.

Reviewer #2:

Q: What is responsible for the difference between the quadratic model and the linear models. How much does the main result say about the advantages of a nonlinear NN over a linearized NN or a kernel model?

A: Our main results show that the quadratic model achieves $\widetilde{O}(d^{\lceil p/2 \rceil})$ sample complexity with neural representations, while the linearized model / kernel suffers from at least $\Omega(d^p)$. The key thing behind is that the generalization performance of the quadratic model depends on (and can benefit from) the conditioning of the covariance of the input (Line 143-147). This enables the sample complexity to be reduced when we feed it with an expressive and isotropic feature map. In contrast, linearized models/neural tangent kernels cannot benefit from feature isotropicity, and generalizes at most as well as a kernel, as stated in the lower bound.

Reviewer #3:

Q: How does whitening affect the proof in Section 4.1? Why not also whiten the raw input?

A: The only effect of whitening is to make the features isotropic, which does not change the expressivity of the features (since it is only a linear transformation) but is beneficial to generalization, as discussed in Line 208-217. We also showed that whitening is not the only option — using unwhitened features $\mathbf{g}$ along with a proper data dependent regularizer on $\mathbf{W}$ gives us exactly the same result as whitening the features (Appendix C.5). On the other hand, existing results on raw representations already assumed $\mathbf{x}$ is exactly or nearly isotropic (Line 244 for `NTK-Raw`; Line 196-198 for `Quad-Raw`). Those bounds won't be improved if we further whiten $\mathbf{x}$ to be exactly isotropic.

Q: What properties of your random representations actually make the difference for the sample complexity here?

A: The key thing that allowed a fixed neural representation to be helpful is that the nonlinearities (along with the width) give us strong expressive power. Linear combinations of these fixed neurons can already express high-complexity nonlinear functions, and such expressivity can be used by the top trainable model to reduce the complexity of the function it has to learn itself. In comparison, when we use the raw input, linear combinations of the input is only a linear function, thus all the "heavylifting" is on the top trainable model, causing the sample complexity to be higher. Therefore, our theory shows that lower-layer representations can ease the burden of learning in the upper layers, which we suspect is also the case in practice even when the representation function is trainable as well.

Reviewer #4:

Q: Are our models overparameterized? How would our results change with smaller m.

A: Our model is overparameterized (we chose D, m to be large) for the purpose of approximating the ground truth function and making the optimization landscape nice. However, our choice of D, m does not explicitly depend on n (Line 181), making our model not necessarily wide enough to memorize the training data. In this sense we are not as overparametrized as the memorization regime.

Q: It would be nice for you to discuss how your work relates to other works that construct random features.

A: Prior work e.g. of Rahimi and Recht considered training only the output layer of the network (a_r in our notation), which is effectively a linear model/kernel method. In constrast our model is non-linear (quadratic) in the trainable parameter $\mathbf{W}$ and has different optimization/generalization behaviors from kernel methods.

Q: Further motivations for considering the quadratic Taylor model.

A: As one example apart from the theoretical benefits shown in this paper, empirically the (full) quadratic Taylor model also approximates the training trajectories of standard neural networks better than the linearized model, as shown in Bai et al. 2020 "Taylorized Training".

We appreciate all the above questions and will incorporate these discussions in our next version.