Reviewer #1

a. We actually have presented the average standard deviation (sd) across different noise/corruption rates for each method in the bottom of Figure 3, as well as in pages 23, 24 of the appendix where accuracies with error bars are shown in full (for the sake of saving space, only the average sd's are shown in the main text). We will make this more clear.

b. As you pointed out, the improvement is marginal in `MNIST` ($2 \sim 3\%$ on average). `MNIST` is used for the pedagogical purpose as its topological feature can be seen clearly both by the naked eye and the persistence diagram. In more complex `ORBIT5K`, we achieved significant improvement both in accuracy and variance reduction over baselines, whereas the state-of-the-art classifier PointNet which is not using the topological information is much worse. We conjecture the degree of improvement is related to how much the topological information is important in the data.

c. We appreciate your suggestion on using kernels for persistent homology for comparative purposes. Indeed, the logarithmic transformation for our choice of g_θ in our paper (174) is a special case of Gaussian kernel for the persistence diagram, which didn't bring much difference compared to the affine transformation (172) in our simulations. Moreover, 1) while our method gives an explicit differentiability guarantee (sec 3.3), it wouldn't be straightforward (or at least require substantial extra work) for other kernel featurization methods so they are not directly compatible in our setting, and 2) in ORBIT5k experiment our method appears to beat the best result of PersLay which utilizes flexible kernel functions. That being said, connection to applying kernels for persistence diagrams sounds very interesting; indeed weighted Gaussian kernel has very similar properties to Landscapes. We will elaborate on this in the discussion section.

Reviewer #2

a,g. Computational complexity depends on how LandLayer is used. Computing the DTM is $O(n + m \log n)$ where n is the input size and m is the grid size. Computing persistence diagram is $O(m^{2+\epsilon})$ for any small $\epsilon > 0$ when we choose the simplicial complex K in line 129 to grow linear with respect to the grid size such as cubical complex or alpha complex. Computing the persistence landscape grows linearly with respect to the number of homological features in the persistence diagram, which is the topological complexity of the input and do not necessarily depend on n or m. For our experiments, we consider fixed grids of size 28×28 and 40×40 as in line 649 and 686, so the computation is not heavy. Also, if we put LandLayer only at the beginning of the deep learning model, then LandLayer can be pre-computed and needs not be calculated at every epoch in the training. We will add this discussion in Section 6.

b. As in line 123, h_{top} is the proposed topological layer equivalent to eq.(5). We will make this more clear. Furthermore, Sec 3.3 is to guarantee the differentiability of h_{top}, one of the main contributions of our paper. The automatic differentiations from `tensorflow` or `pytorch` are correct only if the function is *guaranteed to be differentiable*. Hence, checking the differentiability of h_{top} is critical. Also, h_{top} consists of the piecewise linear function λ_k, which is not differentiable at changepoints. The number of changepoints can be large, so automatic differentiation is inefficient and plugging in the explicit formula for the derivatives of λ_k is better, which is done in the source code.

c,f,h. As stated in 22-24, our paper is more focused on proposing a comprehensive methodological framework with theoretical validation. Experiments are to show the enhanced learnability of neural networks with LandLayer under simple setup, not to build the best model for a particular dataset. Although LandLayer can lead to future applied works, those are beyond the scope of the paper. To get the state-of-the-art, one should design the architecture adaptively to the task, after a considerable amount of research on which model to use and where to place LandLayer. In Table 1, the simple CNN+LandLayer still beats the current state-of-the-art, but that's not our main goal of the paper. In a similar vein, since our method is designed to efficiently capture significant topological features, we intentionally used small sample size to verify the efficiency of our method (242-243), which is another benefit of using topological layer; we conjecture for very large sample size, the gap might be shrinking to some extend. Also please see **b** to Reviewer $\#1$.

d,i. As in **c**, the experiment is to verify the enhancement by adding LandLayer but not to beat existing approaches. LandLayer utilizes topological features, which not only are robust to noise but also provide information that is not extractable by existing methods. As in Table 1, even without the noise, the state-of-the-art PointNet is performing badly due to lack of the topological information. Also, as in line 286-287, LandLayer can be placed anywhere in the deep learning network, and hence it can be combined with denoising layer or regularization techniques as well.

e. We just meant to say improving the overall accuracy by adding the proposed LandLayer, as verified in the result.

j. Equation (4) is the direct application of (2) where X_i's are fed into the place of ϖ_i's and Y_i's into the place of X_i's.

k. Regarding reproducibility, we have provided all the details in Section D.9 of the appendix and the full source code too. We do not see any issues with the reproducibility of our experimental result. Can you please elaborate on this?

Reviewer #3

a. As you pointed out, some recent work proposed their topological layers as in Hofer et al 2017, Carrière et al 2020 (PersLay), etc. But their differentiability is only guaranteed w.r.t the persistence diagram, NOT the layer input. So they can't be placed in the middle of the network, but only in the beginning of the network. To the best of our knowledge, this general differentiability guarantee, not to mention other favorable properties, has not yet appeared in the literature.

b. PersLay used their own persistent homology (i.e., extended diagram) and their own filtration functions, both of which characterize PersLay itself. So we didn't use DTM for PersLay but still reported it for `ORBIT5K` for comparative purpose as it gives the state-of-the-art result. Moreover, for SLayer we used the exactly same DTM filtration. Also see **c** to Reviewer $\#2$.