We'd like to thank all the reviewers for their illuminating reviews. We were pleased to see quite a few remarks of the reviewers highlighting the novelty and strength of our paper, including *a nice and novel contribution to the NTK literature, technical sections are sound, extremely clear and compelling, their approach is sound and well motivated, and is very well written*. Below we address major questions raised by the reviewers and will revise the paper accordingly.

Response to Reviewer #1. R1.1. The goal of this paper. Our ultimate goal is to explain the behaviors of NNs using "NN-simulating" kernels. You are right that LANTK-NTH is a better choice in this direction. However, LANTK-NTH is impractical (footnote 2), so we instead use LANTK-HR in our experiments, because LANTK-HR is similar to LANTK-NTH in both concepts (Sec. 3.1) and experiments (Sec. 4.1).

R1.2. Improvement is small. The improvement of LANTK-HR is limited mainly because we use quite simple higher-order regression methods (line 248-253). A natural implementation of LANTK-HR and LANTK-NTH requires $O(n^4)$ complexity, but our approximation requires only $O(n^2)$. In our experiments, the kernels can improve quite a lot if the second-order component (i.e., $\mathcal{Z}$ in Eq. 5) is good. However, in practice, it's hard to get a good $\mathcal{Z}$ due to the time complexity issue. Still, our simple approximation points out that adding label information can improve NTK.

R1.3. Comparing NTK and LANTK in changing label systems. Good suggestions! Will do this.

R1.4. Relations between LANTK-HR&NTH. You are right that the relation between two LANTKs requires more justification. However, the two numbers still indicates that the "intention" of two LANTKs are similar: they are both trying to increase the similarity between two examples if they come from the same class. We'll add more analysis there.

R1.5. The impact of variance at initialization. The variance indeed comes into play, but we choose to not consider it in our construction because (1) it is not clear to us how to incorporate the variance component in a kernel, as it has no explicitly formula; (2) we did an experiment on fitting kernel regressions using the expected (over the initialization) NTK and the realized NTK, and we found that the former consistently outperforms the latter.

R1.6. On Claim 2.1. We can have good generalization under various label systems, *only if* the data is separable in *both label systems*. Our claim says that there are examples where it's separable in one label system, but not in the other.

Response to Reviewer #2. R2.1. Experimental details. Thanks! We'll provide more details in the revision.

R2.2. The closeness between LANTK-HR and NNs. We disagree on this point. Both empirical evidence and theoretically well-justified design intentions show that LANTK-HR is closer to NN. Particularly, LANTK-HR generalizes better on moderately large datasets like CIFAR-10; (2) it's more locally elastic. This is because its construction is based on aligning with the optimal kernel, which is somewhat independent of the training dynamics (compared to LANTK-NTH which explicitly takes advantage of the training dynamics). The suggestion of trying to prove the trajectory closeness is super interesting and we'll pursue this in the future work.

R2.3. The limited performance of CNN. While the empirical improvement might not look pronounced, we'd like to emphasize that our work is theoretically intended, which echos Reviewer 3's comment *this could open a line of work that eventually gives significant improvement.* Nevertheless, we use current CNN settings for the following reasons: 1) We simply use the default architecture in the tools for NTK computation (Novak et al., 2020). 2) We choose a relatively simple architecture (seven-layer CNNs) mainly because we want to compare it to CNTKs with the same architecture, but the computation cost of deeper CNTKs is huge. 3) The improvement of LANTK is a little limited mainly because of our simple higher-order regression methods (see more in **R1.2**).

R2.4. Use bold y and define $\mathbb{E}_{\text{init}}$. Thanks! We'll revise the paper according to your suggestion.

Response to Reviewer #3. R3.1. Loss inconsistency. The cross-entropy loss in the code is used for *CNN training*, not for NTK/LANTK. For NTK/LANTK, we use kernel regression (implicitly based on least squares loss). For multi-class classification, the label is one-hot (10-dimensional) which is similar to the settings in Arora et al., (2019), and we feel it is acceptable in practice. We choose to work with the least squares loss because: (1) to the best of our knowledge, nearly all of the NTK literature, which we build our results on, work with least squares loss (2) in principle, we can establish an alternative NTK theory based on cross entropy loss (e.g., Ji and Telgarsky 2019 is based on logistic loss), but we won't be able to get closed form formulas for the kernel. Since our focus is to design kernels that simulate NNs and can indeed be computed in practice, we chose to stick to the least squares loss.

R3.2. Concerns about the gradient flow. Similar to the above, the reason we work with continuous dynamics is because: (1) it's analytically simpler; (2) most NTK literature work with gradient flow. We will discuss this point explicitly in the next version. Developing the gradient descent version is left for future work.

Response to Reviewer #4. R4.1. Concerns on test performance for CNN training. This issue only appears in CNN training, and if we use, say, the last iterate performance, the relative improvement of LANTKs will only be better. Nevertheless, we thank the reviewer for pointing this out, and we clearly state this issue in our later version.