Thank you very much for your detailed reviews and comments. In the rebuttal we will focus on the main issue raised: lack of clarity in the description of our theoretical model. At the end of the rebuttal we will address the remaining comments.

Confusion about our landscape model toy task and the definition of n-wedges. A common point you brought up is the difficulty in understanding our loss landscape toy task construction, especially what exactly we mean by the n-wedges. We found that in order to be able to verify whether a particular landscape model matched the behaviour observed in real nets, we needed to implement an explicit simulation.

The simplest version of our toy landscape is constructed as follows. We populate the D-dim weight space with n-dim low-loss attractors we call n-wedges. Each of these n-wedges has n infinitely extended *long* dimensions, and $D - n$ infinitely thin *short* directions. We take each n-tuple of axes, and position a single n-wedge such that its long directions are aligned with them. We then define a surrogate loss $\mathcal{L}_{\text{toy}}(\vec{P} \in \mathbb{R}^D)$ for a network configuration P in this weight space, which we choose to depend monotonically on the L_2 distance to the nearest n-wedge. Luckily for us, this distance is simply $d(P) = \sqrt{\text{sum}\left(\text{sorted}(P)[: D - n]^2\right)}$ – an easy to understand explicit expression

While this construction is very specific, we find that it is the dimensions D and n that influence our results, rather than the specific angles between the n-sheets or their axis-alignment. As such, our toy model serves us well, albeit it doesn't capture many other features of the loss landscape. Nonetheless, on this landscape, we are able to perform connectivity experiments, as well as experiments with optimizing on random hyperplanes, and empirically verify the similarity to real network experiments.

In real nets, we find a large number of weight-space directions in which we can move very far, while the loss doesn't change – those would be the n long directions of the wedge; we also find a small number of extremely sensitive directions in which a small motion incurs a high loss cost – those are the $D - n$ short directions. Together, these define locally an n-dimensional hyperplane of finite thickness in the remaining $D - n$ thin direction, i.e. a *cuboid*. Experimentally we notice a strong effect of radius $r^2 = \sum_i w_i^2$, the sum of squares of all weights. While locally a cuboid, we find that individual parts of the manifold of low loss points radiate from the origin at a well-defined range of angles, like a *wedge*. We find the full low-loss manifold to be a union of those in different directions and orientations. We will include this extended discussion in the paper. We will also include an Appendix with a detailed description of the toy landscape + the code that we use to experiment with it + we will publish a demo Jupyter Notebook / Colab.

R1: More experiments, larger networks, and harder datasets. To strengthen the case for our landscape model, we extended the experiments in our paper to include fully-connected as well as convolutional networks of various sizes (width, depth, non-linearity) including large models such as the **ResNet20v1** (>90% test on CIFAR-10), trained on MNIST, Fashion MNIST and CIFAR-10 & 100. To go beyond classification, we also looked at CNN-based autoencoders. In all cases the results supported our landscape model and we will include them in the final version. This also demonstrates that our landscape model did not overfit to a small CNN on F-MNIST, as it holds for other architectures and datasets.

R5: Overfitting the landscape model to a particular task? New predictions and their empirical observation to the rescue. We constructed a model for the loss landscape of neural networks based on existing observations in literature and our own verification of them. While we were very happy that our model incorporates them all (people in general had trouble reconciling them together), what gave us confidence were new effects that we predicted based on the model, that we only *later* observed in real networks. Those were 1) the existence of $(N - 1)$-dimensional low-loss connectors between N-tuples of independent optima, and the scaling of the number of short (=high curvature) directions in their middle with N, 2) the changing of the predicted labels in the middle of a low-loss connector between two optima, 3) stochastic weight ensembling (SWA) not working when checkpoints are too far from each other (belonging to different wedges). We were aware of none of those at the time of building our model, and only later we predict they should happen, and verified them in real networks.

R2: Getting better at visualizing high-dimensional intuitions in 2D. During the time between the submission and now, we developed a better set of figures and explanations to convey the high-dimensional intuitions in 2D and 3D. For example, we have a better version of Figure 1, where we do not make the wedges circular and smooth, as this was a confusing illustration for some of our readers.

R5: Radial tunnels = what low-dimensional cuts would show. We noticed a confusion about the two types of "tunnels" we discuss: we use the low-loss connectors between two independent optima as an observation to reconcile with our model + a diagnostic tool. The other type of a tunnel – the radial tunnel – is what we would see on 2D cuts through the landscape. At any point in training, making a random, 2D visualization of the loss around our current point, we would (very likely) see a convex depression. As training progresses mainly radially, and at each point there is convex depression around us, we can visualize this as a radial tunnel going out. We will be clearer with the distinction in the final version.