We would like to thank the reviewers for their valuable feedback, which we will duly consider and integrate in our revised manuscript. According to reviewers, our paper presents "a completely novel way to study DNNs and uncovers new structure in their behaviour". Namely, we introduce the *new* notion of Neural Anisotropy Directions (NADs) "which are directions in the input space for which a CNN is biased to be able to easily separate". "What's more, we also provide constructive and efficient ways to find NADs" and validate their role on generalization on complex datasets. "This is an important addition to our understanding of DNNs" with "important insights for the NeurIPS/AI community".

NADs and generalization (R3) *The poison experiment, indeed, does not say much about the alignment of the NADs of CNNs with the features of CIFAR10, and the same behaviour is expected on other datasets*. The main insight in the poison experiment comes from the gradual increase in accuracy when the carrier is placed along higher NADs. Indeed, at the two extremes (first and last NADs) the DNN either picks the carrier or the CIFAR10 features. However, when the carrier is placed at the lower-middle NADs, the continuous increase in accuracy can only be explained if progressively more features from the CIFAR10 data are exploited by the DNN. We do not intend to imply that these features are linear, but that these (possibly non-linear) features can only depend on the NADs before the carrier, which highlights *the existence of an ordered preference of features for DNNs*.

Nevertheless, the experiment in Fig. 8 is precisely intended to demonstrate the alignment of NADs with the features of CIFAR10. In fact, flipping, i.e., reversing *all* the components of each image in the NAD basis, dramatically decreases the performance of these DNNs. Hence, since the flip is a lossless transformation, the only possible explanation is that the inductive bias of the DNN does not align properly with this new representation.

Despite the importance of analyzing the NADs for trained DNNs, the methodology proposed by R3 will unfortunately not work on this setup: The energy of CIFAR10 images is mostly concentrated on the first few NADs, and hence virtually no sample has small components in the first NADs. We are currently studying other avenues of research to test the dependence of trained DNNs on NADs.

NADs beyond pooling (R2,R4) Understanding the role of different architectural components in the shape of NADs is one of our principle active lines of research. In this regard, our preliminary experiments have identified that, beyond pooling and downsampling modules (e.g., striding), padding, skip connections and kernel sizes, can also heavily influence the qualitative behaviour of NADs. We plan to release a complete analysis of this problem in the near future.

NADs and NAS (R2,R4) We see Neural Architecture Search (NAS) as a longrun future application of our ideas, with two main flavours: The introduction of human perception priors, like a frequency or color preference profile; or the distillation of biases from one architecture to another, e.g., for network compression. The main challenge to use NADs on NAS is finding the way to introduce this prior in the optimization. A possible idea is to add a term in the objective that penalizes deviations from the target NADs.

Other datasets for poison experiment (R1,R3,R4) The main computational bottleneck for our validations on real data comes from the need to retrain a full DNN every time we poison the dataset. Unfortunately, the computational budget available to us does not allow to retrain a larger dataset that many times. We hope that the research community will be able to replicate our findings in these benchmarks as our code will be open sourced (R1).

Poison perceptibility (R3) The poison experiment is not designed to be imperceptible, but to test the preference of a DNN for different features. Our rough inspection suggests that perceptibility is influenced by image content, channel, and NAD index. In general, poisoning the first NADs seems more perceptible, but *does not change the image semantics*.

Connection between Sec. 2 and 3 (R1) In Sec. 2, we provide a theoretical analysis of a simple model to illustrate how a particular layer can cause neural anisotropies. This is just an explanatory example, and does not provide a constructive way to identify NADs on a full architecture with multiple layers. Sec. 3 provides a tractable algorithm to compute NADs without explicitly requiring to know the mechanisms that generate them. Due to its complexity, we decided to leave the study of the complex interaction between layers for future work. We thank the reviewer for the suggestion.

NADs and optimization (R1,R2) NADs are not linked to a network instance or a specific training stage. Thus, they do not change during training. Instead, *they shape the optimization landscape and hence influence the training dynamics of an architecture*: When fitting a boundary on "good NADs" DNNs can do this faster than when fitting it on "bad NADs" and this can be seen in the speed of convergence of a DNN on different dirs. (see Fig.2 and Sec. D.3.1. in appendix).

NADs computation (R1,R3) Our method to identify NADs boils down to drawing multiple randomly initialized networks and performing a spectral decomposition on the resulting gradients. Because NADs specify a basis of the input space they have the same dimensionality as the input (e.g., 1024 for 32×32 pixels). The NAD index refers to the index of eigenvectors (R1). Overall, complexity is not an issue (R3), our method is in general very fast and scales to very large input dimensions. Furthermore, the main properties of interest are found on the first few NADs (recall the fast decay of eigenvalues), which are faster to compute. Finally, the number of classes at the final layer is not important: NADs are computed by replacing the final layer with a single output at which we compute the input gradient.