We are grateful to the reviewers for their time and their thoughtful comments, which we believe will improve the paper. We first clarify the comparison with DivideMix and then address all individual comments below. DivideMix is an impressively engineered combination of **multiple different techniques** with associated hyperparameters: a warm-up period, a mixture model, two networks trained with multiple augmentations, a sharpening function, and a confidence penalty. In contrast we propose a **novel regularization term**, grounded in our theoretical analysis, which achieves strong performance **on its own**. Due to its complexity DivideMix takes 2 to 4 times longer to train than our methods (e.g. 1.1h for ELR, 2.3h for ELR+ vs 5.4h for DivideMix for CIFAR-10 on a single Nvidia v100 GPU). Moreover, the results reported for DivideMix on CIFAR10 are obtained by monitoring accuracy on the validation set during training and choosing the highest value, i.e. **not on completely held-out data**. In contrast, we use 10% of the training set for validation, and treat the validation set as a **purely held-out test set** (this also means that we train on less data). Following the same approach as DivideMix, the accuracy for ELR+ is the same or even higher (e.g. 94.6%, 93.5%, 76.5% for 50%, 80%, 90% noise levels vs 94.6%, 93.2%, 76% for DivideMix) but we were not comfortable reporting best validation accuracy, which may overfit the validation set. If we further combine ELR+ with the *unsupervised loss component* in Ref.[20] we outperform DivideMix clearly at high noise levels (94.2%, 79.8% for 80%, 90% respectively), achieving **state-of-the-art performance**. However our focus is to propose a **new tool**, not to combine many different ones to maximize performance. Despite this, ELR+ slightly outperforms DivideMix on Clothing1M and WebVision. We believe that the superior performance of DivideMix on ILSVRC12 is due to the use of semi-supervised learning techniques. We will explain all of this in our revision, including possible limitations as suggested by Reviewer 3.

Reviewer 1 • *The memorization effect is not new to the community.* - We agree. Our contribution is to show that this effect is actually a **fundamental phenomenon in high-dimensional classification** that does not only exist in nonlinear models (e.g. neural networks) and provide a **rigorous mathematical characterization** on linear models.

• *. . . theoretical justification in terms of why co-training and weight averaging can improve results* - This is a great suggestion. We believe that it may be due to a reduction in confirmation bias. We will explain this in the paper.

Reviewer 2 We thank the reviewer for the very careful examination of the proofs for Section 2, and for drawing our attention to several points that should be explained more clearly.

• *In Line 440. . . , the proof assumes that $\theta_t^\top v < .1$. . . In Line 457, the authors make the opposite assumption [$\theta_T^\top \mathbf{v} \geq .1$]* - This is a crucial point: T is defined to be the first time at which $\theta_T^\top \mathbf{v} \geq .1$, i.e. the **end of the early-learning stage** during which the weights become aligned with the true direction (Prop.3). At that point, enough correlation is achieved to produce accurate predictions (Theorem 4). However, at the same time the gradients from the true labels decrease (Prop.5), which eventually leads to memorization (Theorem 6). We will explain this more clearly.

• *both spheres are sufficiently tiny (i.e. all instances are almost identical)* - Surprisingly and counter-intuitively the spheres are **not tiny** due to high-dimensional geometry. The clusters are, approximately, two spheres whose centers are 2 units apart and whose radii are $\sigma\sqrt{p}$, where p is the dimension. When p is large, $\sigma\sqrt{p} \gg 2$, so the spheres are huge compared to the distance between the two classes. We will clarify this important point in our revision.

• *If $\Delta > 1/2$...the majority of labels are flipped* - As defined in our paper, with probability Δ, each label is replaced by a **random** label, not the wrong label. Therefore a $1 - (\Delta/2)$ proportion of the examples are correctly labeled, and the majority of labels are correct for all $\Delta < 1$. We will make this clear.

• *n is very close to the problem dimension [p]* - In high-dimensional statistics it is common to focus on the case that n and p are of the same order. We can always assume that $p \leq n$ by restricting our attention to the n-dimensional span of the training data. If $p \ll n$, then we are in the "low-dimensional" regime, where many of the surprising features of modern ML (such as the early-learning and memorization phenomena we investigate) are known not to occur.

• *the accuracy within the noisy set ... goes up initially before it drops ... not observed in Figure A1 and B1 for linear models* - A close look at the first few iterations of the top right graph in Fig. A1 and B1 shows that there is a decreasing trend of the accuracy for the linear model as well, it is just faster than in the nonlinear model (probably because the linear model does not need to learn complex features). We will adjust the scale of x axis so that this can be seen better.

• *Some of the figures contain redundant information* - We agree and will update the figures.

• *I don't think including the result on logistic regression adds value* - The method is inspired by the observation that the gradient of the deep-learning loss decouples into two parts, one of which is identical to the one in logistic regression. We believe that a rigorous analysis is therefore valuable, and may be helpful to some readers (e.g. Reviewer 4).

• *it is not clear if the improvement is due to the regularization term or ... due to the new target* - This is an excellent point! We will add a graph showing that the accuracy of the targets without regularization is significantly lower (e.g. 77.43% vs 86.12% for CIFAR10 60% noise).

Reviewer 3 • *There are too many hyper-parameters to consider . . . sensitive to λ* - We have 2 parameters for ELR and 4 parameters for ELR+ (DivideMix has 6), and they do not need to be heavily tuned. As mentioned in Section F.3, our results on CIFAR-10, Clothing-1M, and Webvision are obtained with the **exact same** hyper-parameter values (it is possible that tuning them would further increase performance). It is natural for there to be a dependence on λ, but $0.3 \leq \lambda \leq 0.5$ achieves essentially the same performance on CIFAR-10, as shown in Fig. G.1.

Reviewer 4 •*. . . target vector scales with the size of the dataset* - This is correct, but one can store the target vector on disk (along with the labels) and load it whenever required.