We thank the reviewers for their feedback. Our 'formulation is generic and task-agnostic and therefore has the potential for broad applicability' (**R4**). 'The main strength is the robustness of the results and the simplicity of the method' (**R2**). 'The model simplifies existing work' (**R1**) and 'has been applied to many loss functions and tasks without any change in the training' (**R1**). 'The experiments cover different tasks and benchmark datasets' (**R3**).

'It is misleading to claim that the paper is the first work using task-agnostic weights that do not require iterative learning' (R3.2). *We do not make such a claim.* What we claim is: in contrast to SPL variants and any other related work, our approach applies directly to any baseline without any change whatsoever in the training procedure other than plugging the SuperLoss on top (lines 51-57). We believe a simple and easy-to-use idea has potential for great impact.

Motivation for Equation 1 (R3.1). 'Missing a probabilistic explanation' (R2). While it is true that our SuperLoss is not derived from probabilistic considerations, we point out that our approach for designing the SuperLoss *is* principled and insightful for future work. Namely, we establish a connection between curriculum learning and a family of loss functions that we denote as confidence-aware (CA). We review (in Section 2.1 and Section 1 from the supplementary) existing CA losses and study for the first time their properties, in particular their gradient monotonicity (*i.e.* the gradient of the loss *w.r.t.* the network parameters monotonously increases with the confidence) which is at the root of their connection to dynamic curriculum learning. At the same time, we also emphasize their different shortcomings: unfortunately none of the existing CA losses is task-agnostic (Suppl. Section 1). We therefore propose in Section 2.2 the first *generic* CA formulation, *i.e.* that is jointly able to (1) handle losses of any scale (*i.e.* it is homogeneous, see Suppl. Section 2.3); (2) handle both positive- and negative-valued losses (which justifies the squared regularizer log term[1]); and (3) generalize the input loss (i.e. $\mathrm{SL}^*(\ell) \to \ell$ when $\lambda \to \infty$, Suppl. Section 2.2). Our SuperLoss is also easily interpretable since σ^* directly corresponds to the sample weights (Suppl. Section 1). On top of that, our formulation is among the simplest possible ones that achieves all these properties simultaneously. We agree that the exposition of the design process of the SuperLoss can be improved and will update Section 2.2 and the rest of the paper accordingly.

'Does not brings notably new criteria in determining the sample weights' (R3.3). Compared to other methods that learn sample weights via backpropagation in a unified framework like [47], our formulation goes one step further and directly uses the converged value of the weights according to the current state. We thereby avoid additional parameters, hyper-parameters and inconsistencies due to delay (Section 2.3 and Figure 3 left). We obtain better results in fair settings compared to learning the weights via backpropagation (comparison with DataParameters in Table 4 from Suppl.).

'SuperLoss does not show an advantage on clean data' (R3.4). On clean data, the difference is indeed marginal, as for other curriculum learning methods on such standard datasets with recent deep residual networks.

'In noisy label setting [...] its curve overlaps with several baselines' (R3.5). 'The model does not considerably improve performance metrics and is often on par with other approaches' (R1). Other methods are often limited to a single loss like cross-entropy while our SuperLoss applies to any loss and is easy-to-use (about 10 lines of new code).

'DNN can be easily overconfident on wrongly-labeled data (R3.6)'. That is true. However, we observe in practice that noisy samples are well separated from clean samples even under heavy noise, see Figure 6 where noisy samples are clearly downweighted at the end of the training. We will study the extent of this resilience more precisely in the paper.

Large-scale experiments (R4). For classification, WebVision is larger than ImageNet (2.4M images) and additionally contains some noise due to the automatic collection process. We also provide retrieval experiments on Landmarks, where for the first time we show that training a model on the full dataset (160K images) perform better than training on a subset of automatically verified images. This is a strong large-scale result which shows that large datasets automatically collected can be used instead of manually annotated ones.

'The definition of hard and easy examples is limited to their respective confidence scores' (R1). We indeed follow related work in this definition. We are not aware of any large-scale dataset with annotations for the difficulty of samples, and we point that human annotations might actually not correspond to the actual difficulty from a deep model perspective. We separate hard from easy samples at epoch 20, which is already quite significant. About Figure 2, our SuperLoss does not use the full spectrum of the plot as the optimal confidence value σ^* depends on the input loss, thus avoiding samples with low loss (correct prediction) and low confidence by design.

'The paper lacks experiments illustrating turnaround training times of competing models.' (R1) In practice, the observed overhead in training time for SuperLoss on CIFAR100 is 0.4% longer than for the corresponding baseline.

'σ^* treated as constant' (R2). We confirm that and will make it clear.

'The paper partially illustrates a potential challenge in the current state-of-the-art technique but does not elaborate on it' (R1). Thanks for the suggestion, we agree it would be interesting to look deeper into this direction.

[1]For negative-valued losses $\ell < 0$, the probabilistic formulation from [21] $(\ell/\sigma + \log\sigma)$ blows up to $-\infty$ when $\sigma \to 0$ whereas $\ell/\sigma + (\log\sigma)^2$ (our SuperLoss) behaves suitably, see Section 1 and the closed-form solution in Section 2.1 from the supplementary.