We thank the reviewers for their meaningful and valuable comments, which help to improve the quality of our work.

Training with few errors [R1, R2, R3]: Given the small number of errors available to train ConfidNet due to deep neural network (DNN) over-fitting, one common suggestion from reviewers is to use hold-out data. We performed preliminary experiments of this variant at submission time and they were not conclusive. We report here a consolidated evaluation on all datasets to fulfill reviewer's request, shown in the table below for validation sets with 10% of samples. We observe a general performance drop when using a validation set for training TCP confidence. The drop is especially pronounced for small datasets (MNIST), where models reach > 97% train and val accuracies. Consequently, with a high accuracy and a small validation set, we do not get a larger absolute number of errors using val set compared to train set. One solution would be to increase validation set size but this would damage model's prediction performance. By contrast, we take care with our approach to base our confidence estimation on models with levels of test predictive performance that are similar to those of baselines (R2), and on a par with those reported in other papers, *e.g.* Trust Score (ref. [15] in submission). On CIFAR-100, the gap between train accuracy and val accuracy is substantial (95.56% vs. 65.96%), which may explain the slight improvement for confidence estimation using val set. We think that training ConfidNet on val set with models reporting low/middle test accuracies could improve the approach. We would be glad to add this discussion in the paper if accepted. Note that, in discussed future work, we also consider the use of adversarial attacks, image corruption or label noise to generate additional errors to train from.

AUPR-Error (%)	**MNIST** MLP	**MNIST** SmallConvNet	**SVHN** SmallConvNet	**CIFAR-10** VGG-16	**CIFAR-100** VGG-16	**CamVid** SegNet
ConfidNet (using train set)	57.34%	43.94%	50.72%	49.94%	73.68%	50.28%
ConfidNet (using val set)	33.41%	34.22%	47.96%	48.93%	73.85%	50.15%

Positioning of the approach [R1, R2]: We thank R1 and R2 for bringing to our attention related papers on confidence estimation, we will update references accordingly. R1 mentions the use of bi-directional lattice RNN specifically designed for confidence estimation in speech recognition, whereas ConfidNet offers a model- and task-agnostic approach which can be plugged into any DNN. R2: One of the approaches from Blatz *et al.*'04 is similar to our BCE baseline but is not dedicated to training DNNs. DeVries & Taylor'18 work differs from ours since they perform joint training of confidence and classification for out-of-distribution detection (l. 166-169 in our paper). In addition, they use predicted confidence score to interpolate output probabilities and target whereas we specifically defined TCP, a criterion suited for failure prediction. Finally, post-hoc selective classification methods (R2: Gefman & El-Yaniv'17) identify a threshold over a confidence-rate function (*e.g.*, MCP) to satisfy a user-specified risk level, whereas we focus on relative metrics. The approach is compatible with ours and we consider integrating ConfidNet as confidence-rate function in future work.
Comparison with additional baselines [R2]: As suggested, we have implemented the approach of DeVries and Taylor, using their code, for an additional comparison on CIFAR-10 and CIFAR-100. This method obtains resp. 46.07% and 71.16% on AUPR-Error, similar to other baselines but below ConfidNet (49.94% and 73.68%). This confirms that their approach is not specifically designed for failure prediction, unlike ours. Results for all datasets will be reported in Table 1 in the paper. Following R2's suggestions, we will also add coverage-accuracy graphs in supplementary.
Biasing ConfidNet towards misclassifications [R1]: We have performed additional experiments for training ConfidNet with a weighted loss between erroneous and correct predictions, which will be added to supplementary. While ConfidNet trained with BCE presents small improvements, it does not improve TCP regression. Including an instance-based weighting scheme using TCP confidence for training would be an interesting direction for future work.
Improved performances of learning TCP over BCE approach [R2]: In our setting, ConfidNet is trained to match TCP criterion thanks to a regression loss. We have the intuition that TCP regularizes training by providing more fine-grained information about the quality of the classifier regarding a sample's prediction. This is particularly useful in difficult learning cases where there are only few error samples in training set.
Reproducibility [R2]: We will provide link to a GitHub repository with the code and add more implementation details (hyperparameters, train/val split, accuracies) in supplementary to facilitate reproducibility.
Effect on calibration [R3]: Following R3's suggestion, we have studied the effect of our approach on calibration. On CIFAR-100, it turns out that ConfidNet improves calibration over using MCP as confident estimate (15.61% vs. 22.37% on ECE). We have obtained similar results for other datasets. We will include these additional results in supplementary.
Parameters sharing [R3]: Using a network pre-trained for classification indeed reduces computational complexity Besides, we observed that it helps ConfidNet learn most specific layers for confidence estimation (l. 108-110). Hence, this initialization allows a better structuring of the parameter space.
ConfidNet on mismatch conditions [R3]: In case of data distribution shift, performance is likely to drop. Since TCP tends to be less overconfident than MCP on predictions, we expect it to fail more "graciously", though it will eventually suffer like the main classification branch it is attached to. Leveraging dedicated domain adaptation techniques might help to overcome the problem, which is an interesting direction for future work.