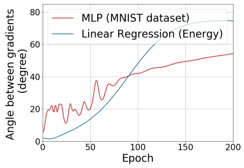
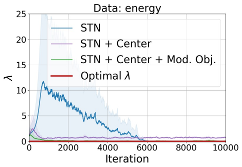
We thank the reviewers for their detailed reviews and constructive feedbacks. We are encouraged that the reviewers found our paper well-written & easy to follow [**R3 R4 R6**], theoretically motivated [**R3 R6**], empirically convincing [**R3 R4 R6**], and novel [**R3**]. The major concerns and questions are addressed below; we will incorporate all suggestions in the next revision and improve the notation (also include a table of notation) as suggested by **R5**.





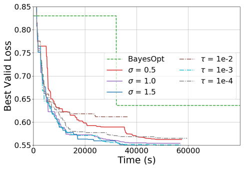
[**R3 R5**] **Undesirable bilevel dynamics for uncentered parameterization.** In support of our claim that the training and validation gradients are well aligned early in training, the left-hand figure shows (for linear regression and an MLP) that the angle between the training and validation gradients starts close to 0 and increases throughout training. Based on the analysis in Section 3.1 (line 152), this alignment causes "spikes" in the hyperparameter values (Fig. 2 on page 7 and Fig. 6 on page 17), even with smaller learning rates. The right-hand figure decouples the centered parameterization from all other changes to the training objective; the centering by itself eliminates the spike, as our analysis predicts. This effect and the poor conditioning of the Gauss-Newton Matrix, limit the optimization performance of uncentered representations.

[**R3**] **Smaller validation gradient ⇒ sensitive to initial hyperparameters?** The uncentered parameterization indeed results in larger hypernetwork gradients, but in a way that is unhelpful for the optimization as described above. Hence, centered parameterizations are *less* sensitive to hyperparameter initializations.

[**R3 R6**] **Applicability to other bilevel problems and limitations of the STN approach.** STNs and Δ-STNs constitute a third general approach to gradient-based bilevel optimization, in addition to implicit differentiation (ID) and unrolling. In relation to those two approaches, (Δ-)STNs are much more efficient, because they amortize the inner optimization and the response Jacobian. The range of applicability is slightly more restricted, as STNs require a single (rather than per-example) inner objective and (like ID but unlike unrolling) require that the outer variables parameterize the training objective. We note that NAS falls into this category and STN approaches can be applied. We will add this discussion to the paper.

[**R5**] **Societal impact.** As almost any application of deep learning involves regularization hyperparameters, and hyperparameter tuning is one stage of a much longer pipeline, any discussion of societal impacts would necessarily be speculative. One predictable impact is to lessen the need for massive computing resources to tune hyperparameters.

[**R3 R4 R6**] **Learning rates and other hyperparameters.** Out of the three aforementioned gradient-based approaches to bilevel optimization — implicit differentiation, unrolling, and (Δ-)STNs — only unrolling is applicable to learning rate tuning. However, even for unrolling, short-horizon objectives suffer from a severe bias [Wu et al, 2018] which makes unrolling impractical for learning rate adaptation unless one uses huge amounts of computation (e.g. [Metz et al., 2019]). Learning rate adaptation remains a hard problem in general. We note that it would be natural to wrap Bayesian optimization (or random search) around the Δ-STN to optimize the remaining hyperparameters that Δ-STNs are inapplicable to. BO suffers from high-dimensional search spaces and pushing the responsibility for regularization hyperparameters onto the Δ-STN can significantly reduce the dimensionality of the BO search space.



[**R3 R4**] **Impact of meta-parameters.** While it is always difficult to rigorously quantify the ease of tuning, we found Δ-STNs to perform well with a default set of meta-parameters (e.g. fixed perturbation scale, choices of α_i) across all tasks we investigated, unlike STNs. Δ-STNs are more forgiving of large perturbation scales than STNs, because they do not need to model the entire response function over the likely range of hyperparameters. Furthermore, Δ-STNs are less sensitive than STNs to initial hyperparameters (e.g. figure 10), for the reasons described in Section 3.1. The right-hand figure shows the effect of different entropy weights, where we grid searched over $\{10^{-2}, 10^{-3}, 10^{-4}\}$ on all experiments. We will provide a more thorough sensitivity analysis in the next revision.

[**R3 R5**] **Validation accuracy, and significance.** Δ-STNs improve over STNs on validation accuracy as well as loss: on AlexNet and VGG16, they improve accuracy by 0.5% and 1.3%, respectively. In Table 1, we showed that Δ-STNs consistently outperform STNs with $4\% \sim 13\%$ improvements in the validation objective across all tasks. We believe that these improvements are significant, since we are using the same set of regularization and data augmentation hyperparameters, without modifying the architecture or introducing new techniques. We also showed that Δ-STNs are more stable and converge faster compared to STNs.

[**R3**] **Scalability.** We modified the statement made on line 249. On a ResNet-32 ImageNet classifier, the Δ-STN requires only a factor of 1.8 more time per epoch compared to training a single network (and this overhead can be made arbitrarily small by updating the hyperparameters less frequently). We did not include ImageNet experiments because, for proper baseline comparisons, it would cost thousands of dollars, but there is no computational obstacle to running Δ-STNs on ImageNet.