Thank you all for the encouraging comments and helpful suggestions. Due to the space limit, we only respond to main concerns, but will incorporate all comments in the final revision. Citations in this rebuttal refer to those in the main paper.

Reviewer 1

- **Bold text**: We bold the highest mean accuracy in all tables and show the bar of our model (σ-reptile) for readability in Fig. 5. We will instead bold the statistically best models in the revision as you suggest.
- **When do we need long adaptation**: We will expand this in the revision. Generally speaking, long adaptation allows higher model capacity for optimization-based meta learning [1-3]. The augmented Omniglot dataset was designed to be challenging for short-horizon optimization, and MAML was found previously not to work for few-shot TTS [6].
- **Selecting modules according to σ and setting to 0 in testing**: This is a very good question. Bayesian approaches always allow some probability of selecting any module. It is common practice in the literature (e.g., Bayesian / regularization-based feature selection [40] and model pruning) to choose features (modules in our case) with large coefficients (σ^2). As explained in Sec D.2 we apply a weak regularizer on σ^2 so that the learned value for task independent modules is close to zero. In practice, we often find that σ^2 of most modules is numerically indistinguishable from 0, which reduces the gap between meta-training and meta-testing (where we set $\sigma = 0$). One exception is the TTS experiment where some pruned modules have non-zero σ^2. All res-blocks have equal size in that case. There, we choose the threshold with the best trade-off between pruned model size and validation performance. Your comment on the relationship of σ^2 with the module size is however worth further investigation. We will add a discussion of this issue.
- **BatchNorm**: Each set of batch-norm parameters (offset, scale) has their own σ. Running stats are not parameters but are computed from inputs. For consistency, all algorithms use transductive learning as in MAML and Reptile.
- **1-step MAML vs σ-Reptile**: MAML requires computing second-order gradients even with 1-inner step.

Reviewer 2 We appreciate your assessment on the strengths of our work.

Reviewer 3

- **Choice of a Normal prior vs sparsity inducing prior**: The shrinkage prior is sparsity inducing for task parameters. Given a weak prior on σ^2, the marginal prior distribution of task parameters is heavy-tailed with tied variance in a group. Similarly, Laplace (spike-and-slab) priors is equivalent to a normal random variable with an exponential (mix of point masses) prior on the variance. The estimate of σ will shrink towards zero if the data does not support task parameters deviating from the prior mean, and thus lead to sparse selection of parameter groups.
- **Showing benefit for some of the aspects that motivated this approach (interpretability, causality, transfer learning or domain adaptation)**: The cited line 30 is a general statement for the purpose of learning resuable modules in the literature. We will clarify that we do not evaluate on all of these aspects. One of our main motivations is for saving space in deployment by sharing the majority of parameters across tasks, which we demonstrate in all experiments. Reviewer 1 also suggests that "*sharing parameters would also allow batching the evaluation of certain layers with a GPU across tasks, which speeds up inference time*". Finally, the discovered middle layers in the WaveNet stack help us better interpret the learned features in the different layers.
- **Bottleneck of hard-coded modules in few-shot domains**: This bottleneck may still exist in few-shot domains. Take TTS for example, training a model from scratch typically requires at least 10K utterances. Learning a model with ~ 100 utterances is a few-shot learning task. Nevertheless, the performance of recent works [5-7] with a hard-coded task-specific component becomes saturated after around 10 utterances. Simply increasing the layer size of the task specific component does not fix the problem.
- **Requiring large number of adaptation steps (L) in TTS**: We will clarify the reference to [6] of this requirement in TTS. Generally speaking, allowing more adaptation steps increases the capacity of optimization-based methods [1-3].
- **Comparison to methods that learn learning rates**: We discuss the relationship of our method with those approaches in Related Work and Appendix C. In Appendix C, we also compare with meta-SGD on a small synthetic example and demonstrate the different behaviors for long adaptation. For real-data experiments, meta-SGD/-Curvature require back-propagation through gradients and thus do not scale to long-adaptation regimes. Finally, SGD performs much worse than Adam for the WaveNet model. It is not clear how to combine meta-SGD/-Curvature with Adam for a fair comparison.
- **Out-of-domain generalization in TTS**: We refer to the different size of the task dataset in meta-training and testing. We will use a more accurate word in the revision.

Reviewer 4

- **Different learning rate in different approaches**: We run a random hyper-parameter search with 100 seeds for each algorithm in each experiment and use the best found value from the validation task set. This is to ensure a fair comparison. Particularly, we find that the reported hyper-parameters in the original papers (MAML, iMAML, Reptile) are not optimal in our implementation. This small difference in learning rate makes essentially no difference in performance.
- **Small scale networks** The WaveNet model in the TTS experiment is a large network with 30 residual blocks. Every block is a dilated and gated causal-convolutional net (see architecture in Figure 12 of Appendix E.2). There are 3M parameters in total. In contrast to most other few-shot meta-learning works that use a pretrained residual network as backbone and meta-learn only the last few layers, we meta-learn the entire CNN stack.