We thank all the reviewers for helpful feedback. We will do our best to answer the reviewers' questions and concerns. Because we could not address all the issues due to the lack of space, we will try to include them in the final version.

[R2, R4] Additional experiments: As R2 and R4 suggested, we show additional experiments for Meta-Dataset (Table A), 20-way classification for a 4-CONV base-learner on miniImageNet (Table B(a)), and regression on randomly sampled sinusoids as in MAML [7] (Table B(b)). Consistent improvements over MAML across a broad set of experiments clearly validate the generalization capability of our method. Following the suggestion of R2, we also perform further ablation study on the input to the hyperparameter generator network, g_ϕ (Table B(c)). We will include the experimental results with more detailed analysis in the final version of the paper.

Table A: Meta-Dataset test accuracy (%) of fo-MAML vs ALFA+fo-MAML (Ours) that are trained on ILSVRC-2012

Model	ILSVRC	Omniglot	Aircraft	Birds	Textures	Quick Draw	Fungi	VGG Flower	Traffic signs	MSCOCO
fo-MAML †	36.09±1.01	38.67±1.39	34.50±0.90	49.10±1.18	56.50±0.80	27.24±1.24	23.50±1.00	66.42±0.96	33.23±1.34	27.52±1.11
Ours	**51.09±1.17**	**67.89±1.43**	**66.34±1.17**	**67.67±1.06**	**65.34±0.95**	**60.53±1.13**	**37.41±1.00**	**84.28±0.97**	**60.86±1.43**	**40.05±1.14**

† Authors of Meta-Dataset [37] use fo-MAML to denote first-order MAML .

Table B: Additional experimental results and the ablation study for input variations

(a) 20-way classification

Model	1-shot (%)	5-shot (%)
MAML	15.21±0.36	18.23±0.39
ALFA+MAML	**22.03±0.41**	**35.33±0.48**

(b) MSE error on regression

Model	5 shots	10 shots	20 shots
MAML	1.24±0.21	0.75±0.15	0.49±0.11
ALFA+MAML	**0.92±0.19**	**0.62±0.16**	**0.32±0.06**

(c) Ablation studies on g_ϕ

Input	5-shot (%)
weight only	68.47±0.46
gradient only	67.98±0.47
weight + grad (ALFA)	**69.12±0.47**

[R1, R5] Differences to prior works: While Meta-SGD [19] and LEO [30] meta-learn inner-loop learning rates, they stay fixed throughout tasks and inner-loop steps during meta-test. In contrast, ALFA learns to generate hyperparameters to adapt the weight-update rule to each task and each step. Although Ravi *et al.*[28] include these adaptive properties, learning weight-update rules directly through their black-box implementation is very difficult, resulting in the limited performance. Instead, ALFA specifies the form of weight-update rule to include the learning rate and weight decay terms that are adaptively generated (Eq. 4), making it practically much more effective. This novel formulation allows ALFA to strike a balance between weight-update with meta-learned but fixed learning rate [19, 30] and direct learning of complex weight-update [28]. We will include clearer discussion with prior works in the updated version of the paper.

[R1, R4] ALFA+Random: Compared to other MAML variants, ALFA+Random requires meta-learning per-parameter weight decay (L136-137) (or learning rate). Note that once the random initialization is given, it should stay fixed throughout meta-train and meta-test altogether (Algorithm 1). Although this increases the total number of parameters for saving the model, the number of *trainable* parameters stays the same as ALFA+MAML. Thus, one interpretation why random initialization works could be that the form of prior knowledge has shifted from initialization (MAML) to weight decay (or learning rate). We have double-checked the reproducibility of the results for random initialization and are confident in the correctness. We will publicly release the code and trained models if our paper gets accepted.

[R2, R4] Claims for Table 4: Note that each column of Table 4 is a separately trained model with different number of inner-loop update steps. Our intended claim was that ALFA+MAML consistently outperforms MAML with 5 steps (Table 1), regardless of the number of steps, even with a single step. We will clarify our explanation in the final version.

[R1] Computational costs: Denoting the number of inner-loop steps as s and the number of layers of f_θ as N, we follow the practices by TADAM [24] to control the range of generated values for stable training, and use $2 \times s \times N$ additional parameters. Including the number of parameters of g_ϕ, ALFA introduces $2sN + 12N^2$ more parameters compared to MAML. For 5-shot 5-way classification with ResNet12 model and 5 inner-loop steps, the average inference time for ALFA+MAML is 645ms per task, which is 3% slower than 628ms for MAML. However, training with ALFA converges faster; ALFA+MAML only required 25 epochs, whereas MAML needed 90 epochs for full convergence.

[R1] Interpretation for negative learning rates: We agree with R1 that negative learning rates can regularize the MAML initialization and reduce meta-overfitting, to which MAML is susceptible [12, 42]. In some sense, this has a similar effect to L2F [2], which dynamically attenuates the initialization. Another possible interpretation could be that negative learning rates prevent overfitting to the support set, since updating the weights with negative learning rates prevents the model from further adaptation to support examples for generalization on unseen examples.

[R5] Input dimension and generalization: The input to g_ϕ is layerwise mean of gradients and weights of f_θ, and hence its dimension is $2N$, where N is **the number of layers of** f_θ (L130-134). The total number of parameters for g_ϕ is $12N^2$ (L142), which is much less than the number of parameters in f_θ (2% for 4-CONV model). Thus, ALFA is easy to generalize to different networks and involves less number of parameters than Meta-SGD [19] and Ravi *et al.* [28].

[R5] Cross-domain explanation: We agree with R5 that our claim in section 4.3.2 is a bit strong. However, as discussed in [4], we believe that the adaptation to novel support examples plays a crucial role in cross-domain few-shot classification when the domain gap between meta-train and meta-test is large (L209-213). Although ALFA is not designed to explicitly learn the domain gap, its adaptive learning capability on new support samples can handle the domain gap to some extent as demonstrated in Table 2. As R5 mentioned, we believe that including the domain variation directly in meta-training can improve the performance, which is an interesting direction for further research. We will address the discussions and reflect them in the final version.