M-L2O: TOWARDS GENERALIZABLE LEARNING-TO-OPTIMIZE BY TEST-TIME FAST SELF-ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning to Optimize (L2O) has drawn increasing attention as it often remarkably accelerates the optimization procedure of complex tasks by "overfitting" specific task type, leading to enhanced performance compared to analytical optimizers. Generally, L2O develops a parameterized optimization method (*i.e.*, "optimizer") by learning from solving sample problems. This data-driven procedure yields L2O that can efficiently solve problems similar to those seen in training, that is, drawn from the same "task distribution". However, such learned optimizers often struggle when new test problems come with a substantially deviation from the training task distribution. This paper investigates a potential solution to this open challenge, by meta-training an L2O optimizer that can perform fast *test-time self-adaptation* to an out-of-distribution task, in only a few steps. We theoretically characterize the generalization of L2O, and further show that our proposed framework (termed as **M-L2O**) provably facilitates rapid task adaptation by locating well-adapted initial points for the optimizer weight. Empirical observations on several classic tasks like LASSO and Quadratic, demonstrate that M-L2O converges significantly faster than vanilla L2O with only 5 steps of adaptation, echoing our theoretical results. All codes will be shared upon acceptance.

1 INTRODUCTION

Deep neural networks are showing overwhelming performance on various tasks, and their tremendous success partly lies in the development of analytical gradient-based optimizers. Such optimizers achieve satisfied convergence on general tasks, with manually-crafted rules. For example, SGD (Ruder, 2016) keeps updating towards the direction of gradients and Momentum (Qian, 1999) follows the smoothed gradient directions. However, these fixed rules disable the analytical optimizers to leverage task-specific information.

Learning to Optimize (L2O), an alternative paradigm emerges recently, aims at *learning* optimization algorithms (usually parameterized by deep neural networks) in a data-driven way, to achieve faster convergence on specific *optimization task* or **optimizee**. Various fields have witnessed the superior performance of these learned **optimizers** compared to analytical optimizers (Cao et al., 2019; Lv et al., 2017; Wichrowska et al., 2017; Chen et al., 2021a; Zheng et al., 2022). Classic L2Os follow a two-stage pipeline: at the *meta-training* stage, L2O trains an optimizer to predict updates for optimizee's parameters by learning from its performance on sample tasks; and at the *meta-testing* stage, the L2O optimizer freezes its parameters and is used to solve new optimizees. In general, L2O optimizers can efficiently solve similar optimizee to those seen during the meta-training stage, or tasks drawn from the same "task distribution".

However, new unseen optimizees could substantially deviate from the training task distribution. As L2O optimizers predict task variables' update based on optimization tasks' dynamics (*e.g.,* gradients), different task distributions may lead to significantly different task dynamics. Therefore, L2O optimizers often incur inferior performance on these distinctive unseen optimizees.

Such challenges have been widely observed and studied in related fields. For example, in meta learning (Finn et al., 2017; Nichol & Schulman, 2018), we aim to enable neural networks to be fast adapted to new tasks with limited samples. In contrastive learning (Chen et al., 2020b; He et al., 2020), we aim to learn an unseen downstream task based on an unsupervised trained embedding. Among all of them, Model-Agnostic Meta Learning (MAML) proposed in Finn et al. (2017), is

one of the most widely-adopted algorithms. Specifically, in the meta-training stage, MAML makes inner-updates for individual tasks, and then make back propagation for aggregated individual task gradients to update meta parameter. This design enablse learned initial to be meta-sensitive to each task, and well-adapted after multiple fine-tuning steps.

Inspired by this, we propose a novel algorithm M-L2O, which incorporates meta adaption design in meta-training L2O stage. Specifically, instead of updating optimizer directly by optimizee gradients, M-L2O introduces extra nested-update for optimizer, and then calculates meta learner updates via meta-updated optimizee gradients. It enforces the optimizer to locate in well-adapted region where few more adaption steps enable optimizer to generalize well on unseen tasks. In summary, the contributions of this paper can be outlined below:

- To address the poor generalization of L2O on out-of-distribution tasks, we propose to incorporate meta adaptation design into L2O training. It enables the learned optimizer to locate in well-adapted initial points, which can be fast adapted to new unseen optimizees after few steps fine-tuning.
- We demonstrate theoretically that our meta adaption design grants M-L2O optimizer faster adaption ability in out of distribution tasks in terms of generalization errors. Our analysis further suggests that a training-like optimizer adaptation task can contribute to better generalization performance. Though such adaptation task selection is different from the common testing-like choice, experimental results validate this claim.
- Extensive experiments consistently demonstrate that in a small number of epochs, the proposed M-L2O enjoys better testing performance compared with different baselines, *e.g.*, vanilla L2O and transfer learning, implying its fast adaptation in practical usage.

2 Related Works

2.1 Learning to Optimize

Learning to Optimize (L2O) captures optimization rules in a data-driven way, and the learned optimizer have demonstrated success on various tasks, including but not limited to black-box (Chen et al., 2017), Bayesian (Cao et al., 2019), minimax optimization problems (Shen et al., 2021), domain adaptation (Chen et al., 2020c; Li et al., 2020), and adversarial training (Jiang et al., 2018; Xiong & Hsieh, 2020). The success of L2O is based on the parameterized optimization rules, which is usually modeled through a long short-term memory network (Andrychowicz et al., 2016), and occasionally as multi-layer perceptrons (Vicol et al., 2021). Although the parameterization is practically successful, it comes with the "curse" of generalization issues. Researchers have established two major direction for improving L2O generalization ability: the first focuses on the generalization to similar optimization tasks but *longer* training iterations. For example, Chen et al. (2020a) customized training procedures with curriculum learning and imitation learning, and Lv et al. (2017); Li et al. (2020) designed rich input features for better generalization. Another direction focuses on the generalization to different optimization tasks: Chen et al. (2021b) studied the generalization for LISTA network on unseen problems, and Chen et al. (2020d) provided theoretical understandings to hybrid deep networks with learned reasoning layers. Our work follows the second trend to theoretically study how traditional L2O and our proposals performs under task distribution shifts.

2.2 Fast Adaptation

Fast adaptation as one of the major goals in meta learning area (Finn et al., 2017; Nichol & Schulman, 2018; Lee & Choi, 2018) which often focuses on generalizing to new tasks with limited samples. MAML (Finn et al., 2017), a famous and effective meta learning algorithm, utilizes the nested-loop for meta adaption. Following this trend, numerous meta learning algorithms chose to compute meta updates more efficiently. For example, FOMAML (Finn et al., 2017) updated only by the first order information, Reptile (Nichol & Schulman, 2018) introduced the extra intermediate variable to avoid Hessian in meta computation, HF-MAML (Fallah et al., 2020) approximated the one-step meta update by Hessian-vector production and Ji et al. (2022) adopted multi-step approximation in updates. Meanwhile, many researchers incorporated various designs to compute meta updates more wisely. For example, ANIL (Raghu et al., 2020) only updated head of network in the inner loop, HSML (Yao et al., 2019) tailored the transferable knowledge to different tasks and MT-net (Lee & Choi, 2018) enabled the meta-learner to learn on each layer's activation space. In terms of theoretical part, (Fallah

et al., 2021) measured the generalization error of MAML theoretically, Fallah et al. (2020) captured the singe inner step MAML convergence rate by Hessian vector approximation. Furthermore, Ji et al. (2022) characterized multiple-steps MAML convergence rate. LFT is recently proposed in Zhao et al. (2022) which combines the meta learning design in Learning to Optimize and demonstrates its better performance for adversarial attack applications.

3 Problem Formulation and Algorithm

In this section, we firstly introduce L2O problem formulation. Then, we propose generalizable self-adapted algorithm M-L2O.

3.1 L2O problem Definition

Most machine learning algorithms adopt analytical optimizer, e.g., SGD to compute parameter updates in general loss function (we call it as **optimizee** or **task**). Instead, L2O aims to estimate such updates by a model, which we call **optimizer**. Specifically, optimizer model takes the optimizee information (e.g. loss, gradient) as input and outputs the update estimation. In this work, we learn optimizer model initial on training tasks, then further update it with adaptation tasks. Finally, we apply such an adapted optimizer model to train the testing optimizees and evaluate its performance.

We define $l(\theta;\xi)$ as the loss function where θ is the **optimizee** parameter, $\xi = \{\xi_j(j = 1, 2, \ldots, N)\}$ denotes the data sample. The empirical and its corresponding population risks are defined as below:

$$\hat{l}(\theta) = \frac{1}{N}\sum_{j=1}^{N} l(\theta;\xi_j), \quad l(\theta) = \mathbb{E}_{\xi} l(\theta;\xi).$$

In L2O, **optimizee** parameter θ is updated by **optimizer**, an update rule parameterized by ϕ and we formulate it as $m(z_t, \phi)$. Specifically, $z_t = z_t(\theta_t;\zeta_t)$ denotes the optimizer input and captures the information (e.g., loss values, gradients) of optimizee with parameter θ_t on data batch ζ_t in t-th iteration. Note that ζ_t refers to a data batch of size N used in t-th iteration while ξ_j refers to the j-th single sample in the data batch. Then, the update rule of θ in t-th iteration is shown as below:

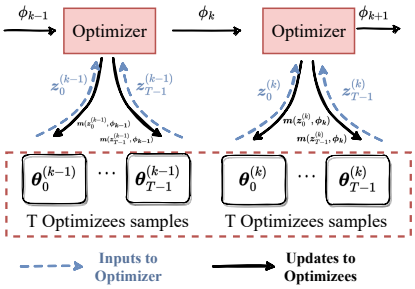


Figure 1: The pipeline of L2O problems.

$$\theta_{t+1}(\phi) = \theta_t(\phi) + m(z_t, \phi). \tag{1}$$

Hence, L2O pipeline is illustrated in Figure 1. In this paper, we only consider using optimizee gradient information for update, i.e., $z_t(\theta_t;\zeta_t) = \nabla_\theta \hat{l}(\theta_t(\phi))$, and then $\nabla_\phi \theta_T(\phi)$ takes a form of

$$\nabla_\phi \theta_T(\phi) = \sum_{i=0}^{T-1}\Big(\prod_{j=i+1}^{T-1}(I + \nabla_1 m(\nabla_\theta \hat{l}(\theta_{T+i-j}), \phi)\nabla_\theta^2 \hat{l}(\theta_{T+i-j}))\nabla_2 m(\nabla_\theta \hat{l}(\theta_i), \phi)\Big), \tag{2}$$

where we assume that ϕ is independent from θ_0 and all samples are independent. The detailed proof is in Appendix Lemma 1.

Since **optimizee** parameter θ is updated by **optimizer** parameterized by ϕ, we further fix the optimizee starting point θ_0 and characterize the loss function w.r.t. optimizer ϕ by introducing the following optimizer empirical and populational risks:

$$\hat{g}_t(\phi) = \hat{l}(\theta_t(\phi)), \quad g_t(\phi) = \mathbb{E}_\xi l(\theta_t(\phi);\xi) = l(\theta_t(\phi)), \tag{3}$$

where $\theta_t(\phi)$ updates in such fashion: $\theta_{t+1}(\phi) = \theta_t(\phi) + m(z_t(\theta_t;\zeta_t), \phi)$. Usually, optimizer ϕ is evaluated and updated after total T-th iteraitons of optimizee, thus the optimal points of optimizer risks in eq. (3) are defined as below:

$$\widetilde{\phi}_* = \arg\min_\phi \hat{g}_T(\phi), \quad \phi_* = \arg\min_\phi g_T(\phi). \tag{4}$$

3.2 M-L2O Algorithm

To improve the poor generalization ability of L2O, we aim at learning a well-adapted optimizer initial by introducing the meta nested-update architecture. Instead of directly adopting ϕ inital, we make one adaptation update firstly. Specifically, we define the new optimizer empirical risk as follows:

$$\hat{G}_T(\phi) = \hat{g}_T(\phi - \alpha\nabla_\phi \hat{g}_T(\phi)), \tag{5}$$

where α is the step size for optimizer inner update. Usually, one optimizee is sampled from multiple distributions and such a design enables optimizer ϕ to locate in a well-adapted balance point for each task. Actually, This nested-update design has been proved successfully in meta learning area, e.g., MAML (Finn et al., 2017). Then, the optimal point of the corresponding meta updated optimizer is:

$$\widetilde{\phi}_{M*} = \arg\min_{\phi} \hat{G}_T(\phi). \tag{6}$$

Based on such meta adapted optimizer loss, we propose M-L2O in Algorithm 1. No that Algorithm 1 only returns the well-adapted optimizer initial point and it requires further finetuning for practical usage in testing loss. To be adapted successfully in new testing optimizee, it is natural to try different optimizees for finetuning. To distinguish between mentioned tasks, we here use different superscripts to denote different tasks, e.g., $g^1(\phi), g^2(\phi), g^3(\phi)$ denote meta training, adaptation and meta testing task individually. Hence, Algorithm 1 returns the meta training output $\widetilde{\phi}^1_{MK}$ and we make its further update on $\hat{g}^2_T(\phi)$. then the testing loss of M-L2O is shown as below:

$$g^3_T(\widetilde{\phi}^1_{MK} - \alpha\nabla_\phi \hat{g}^2_T(\widetilde{\phi}^1_{MK})), \tag{7}$$

where $g^3_T(\phi)$ denotes the meta testing loss.

Algorithm 1 Our Proposed M-L2O.

Input: Inner step size α, Outer learning stepsize β_k, Total epochs K, Epoch number per task S, Optimizer initial point $\widetilde{\phi}^1_{M0}$, Training task $\hat{g}^1$, Adaptation task $\hat{g}^2$, Testing task g^3
for $k = 0, 1, \ldots, K-1$ **do**
  if $\text{mod}(k, S) = 0$: $\theta_0(\widetilde{\phi}^1_{Mk}) = \theta_0$ (random initial) **else** $\theta_0(\widetilde{\phi}^1_{Mk}) = \theta_T(\widetilde{\phi}^1_{M(k-1)})$
  for $t = 0, 1, \ldots, T-1$ **do**
    $\theta_{t+1}(\widetilde{\phi}^1_{Mk}) = \theta_t(\widetilde{\phi}^1_{Mk}) + m(\nabla_\theta \hat{l}_1(\theta_t(\widetilde{\phi}^1_{Mk})))$ Note: $\hat{l}_1(\theta_t(\widetilde{\phi}^1_{Mk})) = \hat{g}^1_t(\widetilde{\phi}^1_{Mk})$
  end for
  Compute $\hat{G}^1_T(\widetilde{\phi}^1_{Mk}) = \hat{g}^1_T(\widetilde{\phi}^1_{Mk} - \alpha\nabla_\phi \hat{g}^1_T(\widetilde{\phi}^1_{Mk}))$
  update: $\widetilde{\phi}^1_{M(k+1)} = \widetilde{\phi}^1_{Mk} - \beta_k \nabla_\phi \hat{G}^1_T(\widetilde{\phi}^1_{Mk})$
end for
Output: $\widetilde{\phi}^1_{MK}$ **Testing Loss:** $g^3_T(\widetilde{\phi}^1_{MK} - \alpha\nabla_\phi \hat{g}^2_T(\widetilde{\phi}^1_{MK}))$

4 Generalization Theorem of M-L2O

In this section, we introduce several assumptions and characterize M-L2O's generalization ability.

4.1 Technical Assumptions and Definitions

Firstly, to characterize the generalization of meta adaptation, it is necessary to make strong convexity assumption which has been widely observed (Li & Yuan, 2017; Du et al., 2019) under over-parameterization condition and adopted on the geometry of functions (Fallah et al., 2021; Finn et al., 2019; Ji et al., 2021).

Assumption 1. *We assume that the function $g^1_T(\phi)$ satisfies $\mu-$strongly convex. This assumption also holds for stochastic $\hat{g}^1_T(\phi)$.*

To capture the relation between L2O function $\hat{g}_T(\phi)$ and M-L2O function $\hat{G}_T(\phi)$, we make following optimal point assumption.

Assumption 2. *We assume there exist non-trivial optimizer optimal point $\widetilde{\phi}^1_{M*}$ which is defined in eq. (6). Non-Trivial means that $\nabla_\phi \hat{g}^1_T(\widetilde{\phi}^1_{M*}) \neq 0$ and $\widetilde{\phi}^1_{M*} \neq \widetilde{\phi}^1_*$. Then, based on $\widetilde{\phi}^1_{M*}$ definition and the existence of trivial solution, for any $\widetilde{\phi}^1_{M*}$, we have following equation:*

$$\widetilde{\phi}^1_* = \widetilde{\phi}^1_{M*} - \alpha\nabla_\phi \hat{g}^1_T(\widetilde{\phi}^1_{M*}),$$

where α is the step size for optimizer inner update. Note that we define the strongly-convex and smoothness of landscape in Assumption 1 and 3 respectively which validate the uniqueness of $\widetilde{\phi}^1_*$.

The above assumption claims that there exist meta optimal points which are different from original optimal point. Such an assumption is natural from experimental view. In MAML experiments, considering there exists a single training task, then we should expect to learn a well-adapted point rather than the task optimal point. Otherwise, MAML will be exactly the same as transfer learning and such a learned point is difficult to transfer when encountering new tasks. Hence, it is natural to make assumption about the existence of non-trivial meta optimal point.

Assumption 3. *We assume Lipschitz properties for all functions $l_i(\theta)(i=1,2,3)$ as follows:*

a) $l_i(\theta)$ *is M-Lipschitz, i.e., for any θ_1 and θ_2, $\|l_i(\theta_1)-l_i(\theta_2)\| \leq M\|\theta_1-\theta_2\|(i=1,2,3)$.*

b) $\nabla l_i(\theta)$ *is L-Lipschitz, i.e., for any θ_1 and θ_2, $\|\nabla l_i(\theta_1)-\nabla l_i(\theta_2)\| \leq L\|\theta_1-\theta_2\|(i=1,2,3)$.*

c) $\nabla^2 l_i(\theta)$ *is ρ-Lipschitz, i.e., for any θ_1 and θ_2, $\|\nabla^2 l_i(\theta_1)-\nabla^2 l_i(\theta_2)\| \leq \rho\|\theta_1-\theta_2\|(i=1,2,3)$.*

d) $m(z,\phi)$ *is M_{m1}-Lipschitz w.r.t. z and M_{m2}-Lipschitz w.r.t. ϕ, i.e.,*

$$\begin{aligned}\|m(z_1,\phi)-m(z_2,\phi)\| &\leq M_{m1}\|z_1-z_2\| \quad \text{for any } z_1 \text{ and } z_2,\\ \|m(z,\phi_1)-m(z,\phi_2)\| &\leq M_{m2}\|\phi_1-\phi_2\| \quad \text{for any } \phi_1 \text{ and } \phi_2.\end{aligned}$$

e) $\nabla^2_\phi \theta_T(\phi)$ *is ρ_θ-Lipschitz, i.e., for any ϕ_1 and ϕ_2, $\|\nabla^2_\phi \theta_T(\phi_1)-\nabla^2_\phi \theta_T(\phi_2)\| \leq \rho_\theta\|\phi_1-\phi_2\|$.*

The above Assumptions (a)(b)(c) also hold for stochastic $\hat{l}_i(\theta)$, $\nabla\hat{l}_i(\theta)$ and $\nabla^2_\theta\hat{l}_i(\theta)(i=1,2,3)$.

The above Lipschitz assumptions are widely adopted in previous optimization works (Fallah et al., 2021; 2020; Ji et al., 2022; Zhao et al., 2022). To characterize the task difference between meta training task and meta testing task, we define Δ_{12} and $\widetilde{\Delta}_{12}$ as follows:

Assumption 4. *We assume there exist union bounds Δ_{12} and $\widetilde{\Delta}_{12}$ to capture the gradient and Hessian differences respectively between meta training task and adaptation task:*

$$\Delta_{12} = \max_\theta \|\nabla_\theta \hat{l}_1(\theta) - \nabla_\theta \hat{l}_2(\theta)\|, \qquad \widetilde{\Delta}_{12} = \max_\theta \|\nabla^2_\theta \hat{l}_1(\theta) - \nabla^2_\theta \hat{l}_2(\theta)\|.$$

Such an assumption has also been made similarly in MAML generalization work (Fallah et al., 2021).

4.2 Main Theorem

In this section, we analyze the generalization error of M-L2O theoretically and compare it with vanilla L2O approach (Chen et al., 2017). Firstly, we characterize the optimizee difference between any two tasks $\nabla_\phi \theta^1_T(\phi)$ and $\nabla_\phi \theta^2_T(\phi)$ in the following form:

Proposition 1. *Based on eq. (2), Assumptions 3 and 4, we obtain*

$$\|\nabla_\phi \theta^1_T(\phi) - \nabla_\phi \theta^2_T(\phi)\| \leq \textstyle\sum_{i=0}^{T-1}(Q^{T-i-1}\Delta_{Ci} + M_{m2}Q^{T-i-2}\sum_{j=i+1}^{T-1}\Delta_{Di}),$$

where $\Delta_{Ci} = \mathcal{O}(Q^i\Delta_{12})$, $\Delta_{Di} = \mathcal{O}(Q^i\Delta_{12} + \widetilde{\Delta}_{12})$ and $Q^i = (1+M_{m1}L)^i$. Furthermore, we characterize task difference of optimizer gradient $\nabla_\phi \hat{g}_T(\phi)$ as follows:

$$\|\nabla_\phi \hat{g}^1_T(\phi) - \nabla_\phi \hat{g}^2_T(\phi)\| = \mathcal{O}(TQ^{T-1}\widetilde{\Delta}_{12} + Q^{2T-1}\Delta_{12}),$$

where $Q = 1 + M_{m1}L$, Δ_{12} and $\widetilde{\Delta}_{12}$ are defined in Assumption 4.

Proposition 1 shows that optimizer gradient landscape difference scales exponentially with the optimizee iteration number T. Specifically, it involves Q^{2T-1} term with gradient difference Δ_{12} and TQ^{T-1} term with Hessian difference $\widetilde{\Delta}_{12}$. Clearly, $Q^{2T-1}\Delta_{12}$ term dominates when T increases which implies optimizee gradient gap is the key component in the optimizer gradient difference.

Theorem 1. *Suppose Assumptions 1, 2, 3 and 4 hold. Considering Algorithm 1 and eq. (7), if we define $\delta_{13} = \|\phi^1_* - \phi^3_*\|$, set $\alpha \leq \min\{\frac{1}{2L}, \frac{\mu}{8\rho_{g_T}M_{g_T}}\}$, $\beta_k = \min(\beta, \frac{8}{\mu(k+1)})$ for $\beta \leq \frac{8}{\mu}$. Then, with probability at least $1-\delta$, we obtain*

$$\begin{aligned}&\mathbb{E}[g^3_T(\widetilde{\phi}^1_{MK} - \alpha\nabla_\phi \hat{g}^2_T(\widetilde{\phi}^1_{MK})) - g^3_T(\phi^3_*)]\\ \leq&(M_{g_T}(1+L_{g_T}\alpha))\left\|\mathcal{O}(1)\frac{M_{g_T}}{\mu^2}\sqrt{\frac{L_{g_T}+\rho_{g_T}\alpha M_{g_T}}{\beta K} + \frac{M_{g_T}}{\beta\sqrt{K}}}\right\| + M_{g_T}\left\|\frac{2\sqrt{2}M_{g_T}}{\mu\sqrt{\delta N}}\right\| + M_{g_T}\delta_{13}\end{aligned}$$

$$+ M_{g_T}\alpha\mathcal{O}(TQ^{T-1}\widetilde{\Delta}_{12} + Q^{2T-1}\Delta_{12}),$$

where $Q = 1 + M_{m1}L$, $M_{g_T} = \mathcal{O}(Q^{T-1})$, $L_{g_T} = \mathcal{O}(TQ^{T-2} + Q^{2T-2})$, $\rho_{g_T} = \mathcal{O}(TQ^{2T-3} + Q^{3T-3})$, *$K$ is total epoch number for meta training, N is the batch size for optimizer training.*

For further understanding of generalization error, we make the following remark firstly.

Remark 1 (The choice of α)**.** *In Theorem 1, we set* $\alpha \leq \frac{\mu}{8\rho_{g_T}M_{g_T}} = \mathcal{O}(\frac{1}{TQ^{3T-4}+Q^{4T-4}})$, *and thus error term* $M_{g_T}\alpha\mathcal{O}(TQ^{T-1}\widetilde{\Delta}_{12}+Q^{2T-1}\Delta_{12})$ *vanishes with larger T. If we fix the iteration number T, then such error term is determined by gradient bound Δ_{12} and Hessian bound $\widetilde{\Delta}_{12}$.*

The key components that lead to Q dependency are the Lipschitz properties to characterize the L2O loss landscape, e.g. the Lipschitz term $L_{gT} = \mathcal{O}(TQ^{T-2} + Q^{2T-2})$ defined for $\nabla_\phi \hat{g}_T(\phi)$. The reason is due to our nested updated procedure $\theta_{t+1}(\phi) = \theta_t(\phi) + m(\nabla_\theta \hat{l}(\theta_t(\phi)), \phi)$. If we wanna take the gradient of last update term $\hat{g}_T(\phi) = \hat{l}(\theta_T(\phi))$ over ϕ, it requires to compute gradients iteratively for all $t = 0, 1, \ldots, T-1$ and thus contains the exponential term.

Then, it can be observed that generalization error of M-L2O contain three parts: (i) The first term determined by $\sqrt{\frac{L_{g_T}+\rho_{g_T}\alpha M_{g_T}}{\beta K}} + \frac{M_{g_T}}{\beta\sqrt{K}}$ is dominated by training epoch K. Such an error term characterizes how meta training influences the generalization. (ii) The second term $\|\frac{2\sqrt{2}M_{g_T}}{\mu\sqrt{\delta N}}\|$ reflects the empirical error introduced by limited sample, and hence controlled by sample size N. (iii) The last two error terms capture task differences. Specifically, δ_{13} measures the training and testing optimal points gap while Δ_{12} and $\widetilde{\Delta}_{12}$, which dominate the last error term and represent the gradient and Hessian union bounds, reflect the geometry difference between training and adaptation tasks.

For better comparison with L2O, we make the following remark about generalization of M-L2O and Transfer Learninig.

Remark 2 (Comparison with Transfer Learning)**.** *We can rewrite M-L2O generalization error in Theorem 1 in the following form:*

$$\begin{aligned}&g_T^3(\widetilde{\phi}_{MK}^1 - \alpha\nabla_\phi \hat{g}_T^2(\widetilde{\phi}_{MK}^1)) - g_T^3(\phi_*^3)\\ &\leq M_{g_T}\|\widetilde{\phi}_{MK}^1 - \widetilde{\phi}_{M*}^1\| + M_{g_T}\|\widetilde{\phi}_*^1 - \phi_*^1\| + M_{g_T}\alpha\|\nabla_\phi \hat{g}_T^2(\widetilde{\phi}_{MK}^1) - \nabla_\phi \hat{g}_T^1(\widetilde{\phi}_{M*}^1)\| + M_{g_T}\delta_{13}.\end{aligned}$$

For L2O Transfer Learning, the generalization error is shown below:

$$\begin{aligned}&g_T^3(\widetilde{\phi}_K^1 - \alpha\nabla_\phi \hat{g}_T^2(\phi)) - g_T^3(\phi_*^3)\\ &\quad\leq M_{g_T}\|\widetilde{\phi}_K^1 - \widetilde{\phi}_{M*}^1\| + M_{g_T}\|\widetilde{\phi}_*^1 - \phi_*^1\| + M_{g_T}\alpha\|\nabla_\phi \hat{g}_T^2(\widetilde{\phi}_K^1) - \nabla_\phi \hat{g}_T^1(\widetilde{\phi}_{M*}^1)\| + M_{g_T}\delta_{13},\end{aligned}$$

where $\delta_{13} = \|\phi_*^1 - \phi_*^3\|$, $\widetilde{\phi}_K^1$ *represents transfer learning L2O learned point after K epochs.*

The generalization error gap between M-L2O and Trasnfer Learning can be categorized into two parts: (i) Difference between $\|\widetilde{\phi}_{MK}^1 - \widetilde{\phi}_{M*}^1\|$ and $\|\widetilde{\phi}_K^1 - \widetilde{\phi}_{M*}^1\|$, (ii) Difference between $\|\nabla_\phi \hat{g}_T^2(\widetilde{\phi}_{MK}^1) - \nabla_\phi \hat{g}_T^1(\widetilde{\phi}_{M*}^1)\|$ and $\|\nabla_\phi \hat{g}_T^2(\widetilde{\phi}_K^1) - \nabla_\phi \hat{g}_T^1(\widetilde{\phi}_{M*}^1)\|$. If we assume $\|\nabla_\phi \hat{g}_T^2(\phi)\| \approx \|\nabla_\phi \hat{g}_T^1(\phi)\|$, then both differences can be characterized by the gap between $\|\widetilde{\phi}_{MK}^1 - \widetilde{\phi}_{M*}^1\|$ and $\|\widetilde{\phi}_K^1 - \widetilde{\phi}_{M*}^1\|$. Since $\widetilde{\phi}_{MK}^1$ is trained to converge to $\widetilde{\phi}_{M*}^1$ as epoch K increases, it is natural to see that M-L2O ($\|\widetilde{\phi}_{MK}^1 - \widetilde{\phi}_{M*}^1\|$) enjoys smaller generalization error compared with Transfer Learning ($\|\widetilde{\phi}_K^1 - \widetilde{\phi}_{M*}^1\|$).

We further compare above theory with previous theoretical works. In the L2O area, Heaton et al. (2020) analyzed the convergence of proposed safe-L2O but not generalization while Chen et al. (2020d) analyzed the generalization of quadratic-based L2O. Instead, we develop the generalization on a general class of L2O problems. In the meta-learning area, the previous works have demonstrated the convergence and generalization of MAML (Ji et al., 2022; Fallah et al., 2021). We firstly apply such MAML results to measure the distance between training learned point and training optimal point. Note that our L2O theory characterizes transferability of learned point on meta testing tasks. Then, we develop the theorem based on both L2O and meta learning results. In conclusion, our theoretical novelty lies in three aspects: ① Firstly theoretically characterize a generic class of L2O generalization. ② Incorporating the MAML results in our meta learning analysis. ③ We theoretically prove that both training-like and testing-like adaptation contribute to better generalization in L2O.

5 EXPERIMENTS

In this section, we list experimental settings in details and present the results, which show a great alignment between our empirical observations and theoretical outcomes.

5.1 EXPERIMENTAL CONFIGURATIONS

Backbones and observations. The L2O backbone we use for all the experiments is a single-layer LSTM network with a hidden dimension of 20. We follow Lv et al. (2017); Chen et al. (2020a) to use the parameters' gradients and its normalized momentum to construct the observation vectors.

Optimizees. We conduct experiments on three different types of optimizees, LASSO, Quadratic and Rosenbrock (Rosenbrock, 1960). The formulation of the Quadratic problem is $\min_{\boldsymbol{x}} \frac{1}{2}\|\mathbf{A}\boldsymbol{x} - \mathbf{b}\|_2$ and the formulation of the LASSO problem is $\min_{\boldsymbol{x}} \frac{1}{2}\|\mathbf{A}\boldsymbol{x} - \mathbf{b}\|_2 + \lambda\|\boldsymbol{x}\|_1$, where $\mathbf{A} \in \mathbb{R}^{d\times d}$, $\mathbf{b} \in \mathbb{R}^d$. We set $\lambda = 0.005$. The exact form of the Rosenbrock problem can be found in Section A.5. At the meta-training/testing stage, optimizees $\xi_{\text{train}}/\xi_{\text{test}}$ are sampled from $D_{\text{train}}/D_{\text{test}}$, where D_{train} and D_{test} are pre-specified distribution(s), respectively. When performing adaptation, we sample optimizees ξ_{adapt} from D_{adapt}.

Baselines and Training Settings. We compare M-L2O against three baselines: (1) *Vanilla L2O*, where we train a randomly initialized L2O optimizer on ξ_{adapt} for only 5 steps; (2) *Transfer Learning* (TL), where we first meta-train an randomly initialized L2O optimizer on ξ_{train}, and then adapt by fine-tuning on ξ_{adapt} for 5 steps; (3) *Direct Transfer* (DT), where we meta-train an randomly initialized L2O optimizer on ξ_{train} only. M-L2O follows a fair setting, *i.e.*, meta-train on $\xi_{\text{train}}$and adapt on the same ξ_{adapt}. The evaluation are conducted on the same set of optimizees for testing (*i.e.*, ξ_{test}), and the minimal ***logarithm*** of objective values on this optimizee will be reported.

In all experiments, we set the number of optimizee iterations T to be 20 when meta-training the L2O optimizers and adapting on optimizees. Actually, in large scale experiments (Zheng et al., 2022; Shen et al., 2021) where neural networks are selected as tasks, a common choice is 5 for T and our choice $T = 20$ is for better experimental performance. The value of total epochs K is set to 5000, and we adopt the curriculum learning technique (Chen et al., 2020a) to adjust the number of epochs S per task. Practically, we use the Adam optimizer (Kingma & Ba, 2014) to update the optimizers' weights (*i.e.,* ϕ) and the learning rate is fixed at 1×10^{-4}.

5.2 FAST ADAPTATION RESULTS OF M-L2O

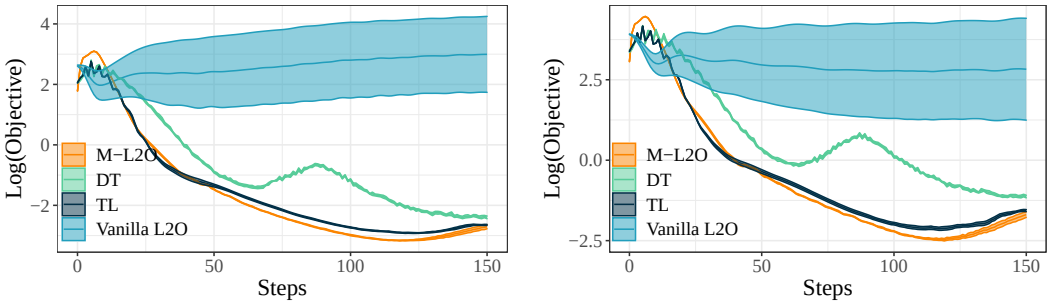
Experiments on LASSO optimizees. We begin with experiments on LASSO optimizees. For ξ_{train}, the coefficient matrix $\mathbf{A}$ is sampled from a mixture of uniform distributions $\{\mathrm{U}(0, 0.1), \mathrm{U}(0, 0.5), \mathrm{U}(0, 1)\}$. For ξ_{test} and ξ_{adapt}, the coefficient matrices $\mathbf{A}$ are sampled from the normal distribution with the standard deviation σ. We provide experiments with $\sigma = 50, 100$ and the results are collected in Figure 2a and 2b, where we show that:

① M-L2O consistently achieves faster convergence speed compared to other baselines methods, validating the effectiveness of our proposals. Vanilla L2O adapts on ξ_{adapt} for only 5 steps from random initialization and therefore yields the worst performance (largest object values).

② Although Direct Transfer (DT) learns optimization rules from training optimizees, the optimizee for testing is substantially different due to larger variance in coefficients thus the learned rules cannot generalize well. The Transfer Learning (TL) baseline's improvement compared to DT highlights the values of adaptation when the test optimizee deviates significantly from those seen in training as the optimizer presumably learns new knowledge during adaptation.

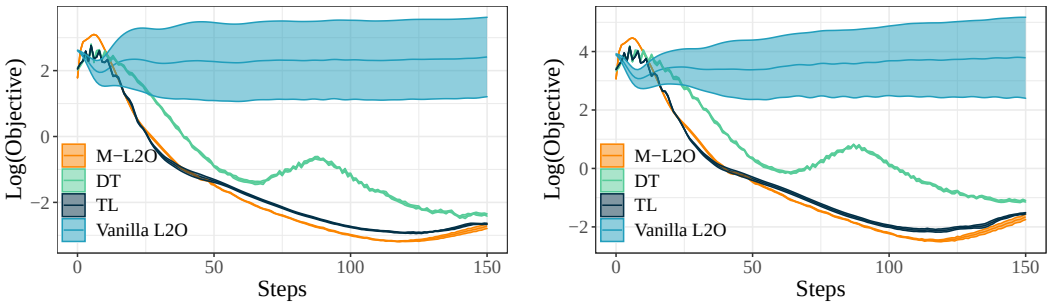
③ Finally, M-L2O shows the best performance overall, reducing the logarithm of the objective values by approximately 0.2 when $\sigma = 50$ and by 1 when $\sigma = 100$, respectively. M-L2O learns a well-adapted initial weights for optimizers, which enables rapid self-adaptation and leads to better performance compared to baselines.

Experiments on Quadratic optimizees. We then evaluate another optimizee, *i.e.*, the Quadratic problem. The optimizees' weight coefficients $\mathbf{A}$ are also sampled from the normal distribution. We conduct two evaluation, where $\sigma = 50, 100$, respectively. The results are shown in Figure 3. We can observe a similar trend.



(a) Performance on $\mathcal{N}(0, 50^2)$ (b) Performance on $\mathcal{N}(0, 100^2)$

Figure 2: Comparison of convergence speeds on target distribution of LASSO optimizees. We repeat the experiments for 10 times and show the 95% confidence intervals are shown in the figures.



(a) Performance on $\mathcal{N}(0, 50^2)$ (b) Performance on $\mathcal{N}(0, 100^2)$

Figure 3: Comparison of convergence speeds on target distribution of Quadratic optimizees. We repeat the experiments for 10 times and show the 95% confidence intervals are shown in the figures.

Experiments on more LASSO optimizees. We continue to vary the choices of the standard deviation σ for distributions to sample the coefficients matrices $\mathbf{A}$ for ξ_{adapt} and ξ_{test}, and record the minimum logarithm of each method's objective value in Table 1. We find that:

① Under small levels of σ, adaptation is not necessarily required (and even unintentionally harmful) for normally trained L2O. M-L2O performs marginally inferior to Transfer Learning (TL), which is presumably due to the high similarity between training and testing tasks. As M-L2O aims at locating a general initial points, L2O optimizers that train directly on similar tasks could generalize well. After adapting on one single task, L2O optimizers forget certain knowledge learned during meta-training, which might be still useful for new but similar tasks.

② However, as the similarity between training and testing tasks is descending (characterized by larger σ) over some "sweet point", M-L2O starts to show considerable advantage. For $\sigma > 50$, M-L2O shows consistent advantages that are greater than 0.1 in terms of the logarithm loss, empirically demonstrating that the located initial weights have enabled fast adaptation to new "out-of-distribution" tasks with large deviation.

Table 1: Minimal logarithm loss of different methods on LASSO under different levels of σ. We report the 95% confidence interval from 10 repeated runs.

σ	Methods			
	Vanilla L2O	M-L2O	DT	TL
10	$0.033_{\pm 0.661}$	$-3.712_{\pm 0.004}$	$\mathbf{-4.233}_{\pm 0.016}$	$-4.077_{\pm 0.015}$
25	$1.559_{\pm 0.789}$	$-3.433_{\pm 0.011}$	$\mathbf{-4.125}_{\pm 0.011}$	$-4.019_{\pm 0.017}$
50	$0.550_{\pm 0.476}$	$\mathbf{-3.285}_{\pm 0.013}$	$-3.098_{\pm 0.021}$	$-3.181_{\pm 0.011}$
100	$2.435_{\pm 1.500}$	$\mathbf{-2.408}_{\pm 0.037}$	$-1.775_{\pm 0.034}$	$-1.961_{\pm 0.050}$
200	$4.104_{\pm 1.300}$	$\mathbf{-1.396}_{\pm 0.035}$	$-0.453_{\pm 0.075}$	$-0.982_{\pm 0.086}$

5.3 ADAPTATION WITH SAMPLES FROM DIFFERENT TASK DISTRIBUTION

In Section 5.2, we constrain the standard deviation of the distribution to sample $\mathbf{A}$ is the same in LASSO optimizee. Actually, such restriction is not necessary in experiments while our theory is developed with different adaptation and testing optimizees. Hence, we vary the distribution where we draw the optimizee ξ_{adapt} for adaptation and its testing results are shown in Figure 4.

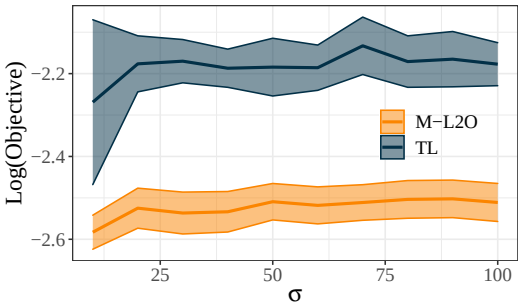


Figure 4: Performance on LASSO optimizees. We vary the standard deviation for sampling the weight matrix $\mathbf{A}$ for the LASSO optimizees. We visualize both the mean and the confidence interval in the figure.

In Theorem 1, we have characterized the generalization of M-L2O with flexible distribution adaptation tasks. Based on the theoretical results, similar geometry landscape (smaller $\Delta_{12}, \widetilde{\Delta}_{12}$) between training task and adaptation task contribute to smaller generalization loss defined in eq. (7). This claim has been validated in our designed experiments in Figure 4. Specifically, we sample σ with smaller values $\{0.1, 0.5, 1\}$ for training task, and sample with larger values $\{50, 100\}$ for testing task. As shown in Figure 4, when we choose smaller training-like adaptation σ (e.g., 10) which implies smaller Δ_{12} and $\widetilde{\Delta}_{12}$, M-L2O enjoy a better testing performance. In conclusion, training-like adaptation task contribute to better generalize performance.

Meanwhile, it is natural to think that the task differences $(\Delta_{23}, \widetilde{\Delta}_{23})$ between adaptation and testing might affect the M-L2O's generalization ability as well since if we enforce optimizer to adapt in testing optimizees, the adapted optimizer should generalize well on testing optimizee. To further understand the relationship between generalization and adaptation-testing task differences, we rewrite M-L2O generalization error in Theorem 1 in the following form with $\widetilde{\phi}_*^3 = \widetilde{\phi}_{M*}^3 - \alpha \nabla_\phi \hat{g}_T^3(\widetilde{\phi}_{M*}^3)$ and $\widetilde{\delta}_{13}^M = \|\widetilde{\phi}_{M*}^1 - \widetilde{\phi}_{M*}^3\|$:

$$\begin{aligned} & g_T^3(\widetilde{\phi}_{MK}^1 - \alpha \nabla_\phi \hat{g}_T^2(\widetilde{\phi}_{MK}^1)) - g_T^3(\phi_*^3) \\ & \leq M_{gT}(\|\widetilde{\phi}_{MK}^1 - \widetilde{\phi}_{M*}^1\| + \|\widetilde{\phi}_*^3 - \phi_*^3\| + \alpha \|\nabla_\phi \hat{g}_T^2(\widetilde{\phi}_{MK}^1) - \nabla_\phi \hat{g}_T^3(\widetilde{\phi}_{M*}^3)\| + \widetilde{\delta}_{13}^M). \end{aligned} \tag{8}$$

In eq. (8), M-L2O generalization error is partly captured by $\|\nabla_\phi \hat{g}_T^2(\widetilde{\phi}_{MK}^1) - \nabla_\phi \hat{g}_T^3(\widetilde{\phi}_{M*}^3)\|$ which is controlled by optimizer difference $\|\nabla_\phi \hat{g}_T^2(\phi) - \nabla_\phi \hat{g}_T^3(\phi)\|$. From Proposition 1, we know that this term is determined by optimizee difference Δ_{23} and $\widetilde{\Delta}_{23}$. Similarly as Theorem 1, we achieve the theoretical claim that smaller testing-adaptation optimizee difference implies better testing performance. This result has been demonstrated in Figure 4 where TL generalizes well with larger testing-like adaptation σ. M-L2O also benefit with larger σ (e.g., $\sigma = 100$) in some cases .

To summarize, both training-like and testing-like adaptation task imply better testing performances. As shown in Figure 4, training-like adaptation implies better generalization in L2O. A possible reason is that when the testing task deviates a lot away the training task, it will be hard for adaptation task to generalize well on such testing task in the limited adaptation steps. In this scenario, the training-like adaptation which provides a handy solution can enjoy considerable performance improvement.

6 CONCLUSION AND DISCUSSION

In this paper, we propose a self-adapted L2O algorithm (M-L2O), which is incorporated with meta adaptation. Such design enables optimizer to locate in well-adapted initial points, facilitating its fast adaptation ability with only a few updates. Our superior generalization performances in out of distribution tasks have been characterized theoretically and validated empirically across various scenarios. Comprehensive empirical results further demonstrate that training-like adaptation task can contribute to better testing generalization which is aligned with theoretical analysis.

In terms of further explorations, one potential direction is to develop the convergence analysis for L2O. L2O involves the delicate architecture which enables the interactions between optimizer and optimizee. Though this architectures has been fully studied and implemented in experiment, we still lack theoretical understanding about its influences towards convergence rates. Furthermore, it will be more interesting to consider meta adaptation in analyzing L2O convergence from theoretical view.

REFERENCES

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems (NeurIPS)*, 2016.

Yue Cao, Tianlong Chen, Zhangyang Wang, and Yang Shen. Learning to optimize in swarms. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 15018–15028, 2019.

Tianlong Chen, Weiyi Zhang, Jingyang Zhou, Shiyu Chang, Sijia Liu, Lisa Amini, and Zhangyang Wang. Training stronger baselines for learning to optimize. *arXiv preprint arXiv:2010.09089*, 2020a.

Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to optimize: A primer and a benchmark. *arXiv preprint arXiv:2103.12828*, 2021a.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020b.

Wuyang Chen, Zhiding Yu, Zhangyang Wang, and Animashree Anandkumar. Automated synthetic-to-real generalization. In *International Conference on Machine Learning (ICML)*, pp. 1746–1756, 2020c.

Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Hyperparameter tuning is all you need for lista. *Advances in Neural Information Processing Systems*, 34:11678–11689, 2021b.

Xinshi Chen, Yufei Zhang, Christoph Reisinger, and Le Song. Understanding deep architecture with reasoning layer. *Advances in Neural Information Processing Systems*, 33:1240–1252, 2020d.

Yutian Chen, Matthew W Hoffman, Sergio Gómez Colmenarejo, Misha Denil, Timothy P Lillicrap, Matt Botvinick, and Nando De Freitas. Learning to learn without gradient descent by gradient descent. In *International Conference on Machine Learning (ICML)*, pp. 748–756, 2017.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Generalization of model-agnostic meta-learning algorithms: Recurring and unseen tasks. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. In *International Conference on Machine Learning (ICML)*, pp. 1920–1930, 2019.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2020.

Howard Heaton, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Safeguarded learned convex optimization. *arXiv preprint arXiv:2003.01880*, 2020.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *International Conference on Machine Learning (ICML)*, pp. 4882–4892, 2021.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Theoretical convergence of multi-step model-agnostic meta-learning. *Journal of Machine Learning Research (JMLR)*, 2022.

Haoming Jiang, Zhehui Chen, Yuyang Shi, Bo Dai, and Tuo Zhao. Learning to defense by learning to attack. *arXiv preprint arXiv:1811.01213*, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. In *International Conference on Machine Learning (ICML)*, 2018.

Chaojian Li, Tianlong Chen, Haoran You, Zhangyang Wang, and Yingyan Lin. Halo: Hardware-aware learning to optimize. In *European Conference on Computer Vision (ECCV)*, pp. 500–518. Springer, 2020.

Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. *Advances in neural information processing systems (NeurIPS)*, 2017.

Kaifeng Lv, Shunhua Jiang, and Jian Li. Learning gradient descent: Better generalization and longer horizons. In *International Conference on Machine Learning (ICML)*, pp. 2247–2255, 2017.

Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1): 145–151, 1999.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In *International Conference on Learning Representations*, 2020.

HoHo Rosenbrock. An automatic method for finding the greatest or least value of a function. *The computer journal*, 3(3):175–184, 1960.

Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *The Journal of Machine Learning Research (JMLR)*, 2010.

Jiayi Shen, Xiaohan Chen, Howard Heaton, Tianlong Chen, Jialin Liu, Wotao Yin, and Zhangyang Wang. Learning a minimax optimizer: A pilot study. In *International Conference on Learning Representations (ICLR)*, 2021.

Laurits Tani, Diana Rand, Christian Veelken, and Mario Kadastik. Evolutionary algorithms for hyperparameter optimization in machine learning for application in high energy physics. *The European Physical Journal C*, 81(2):1–9, 2021.

Paul Vicol, Luke Metz, and Jascha Sohl-Dickstein. Unbiased gradient estimation in unrolled computation graphs with persistent evolution strategies. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10553–10563. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/vicol21a.html.

Olga Wichrowska, Niru Maheswaranathan, Matthew W Hoffman, Sergio Gomez Colmenarejo, Misha Denil, Nando de Freitas, and Jascha Sohl-Dickstein. Learned optimizers that scale and generalize. In *International Conference on Machine Learning (ICML)*, 2017.

Yuanhao Xiong and Cho-Jui Hsieh. Improved adversarial training via learned optimizer, 2020.

Huaxiu Yao, Ying Wei, Junzhou Huang, and Zhenhui Li. Hierarchically structured meta-learning. In *International Conference on Machine Learning (ICML)*, 2019.

Pu Zhao, Parikshit Ram, Songtao Lu, Yuguang Yao, Djallel Bouneffouf, Xue Lin, and Sijia Liu. Learning to generate image source-agnostic universal adversarial perturbations. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2022.

Wenqing Zheng, Tianlong Chen, Ting-Kuei Hu, and Zhangyang Wang. Symbolic learning to optimize: Towards interpretability and scalability. *arXiv preprint arXiv:2203.06578*, 2022.