Enabling Few-Shot Learning with PID Control: A Layer Adaptive Optimizer

Le Yu [*1 2 3] **Xinde Li** [*1 2 3 4] **Pengfei Zhang** [*1 2] **Zhentong Zhang** [1 2] **Fir Dunkin** [1 2]

Abstract

Model-Agnostic Meta-Learning (MAML) and its variants have shown remarkable performance in scenarios characterized by a scarcity of labeled data during the training phase of machine learning models. Despite these successes, MAML-based approaches encounter significant challenges when there is a substantial discrepancy in the distribution of training and testing tasks, resulting in inefficient learning and limited generalization across domains. Inspired by classical proportional-integral-derivative (PID) control theory, this study introduces a Layer-Adaptive PID (LA-PID) Optimizer, a MAML-based optimizer that employs efficient parameter optimization methods to dynamically adjust task-specific PID control gains at each layer of the network, conducting a first-principles analysis of optimal convergence conditions. A series of experiments conducted on four standard benchmark datasets demonstrate the efficacy of the LA-PID optimizer, indicating that LA-PID achieves state-of-the-art performance in few-shot classification and cross-domain tasks, accomplishing these objectives with fewer training steps. *Code is available on https://github.com/yuguopin/LA-PID.*

1. Introduction

Few-shot learning (Aggarwal et al., 2023; Luo et al., 2023; Song et al., 2023) aims to adapt to new tasks by training a new classifier with only a small set of labeled image sample, and even generalizing to unseen query examples. From this research, effectively leveraging prior knowledge and adaptively training a network with limited labeled samples for new tasks is an increasingly significant challenge.

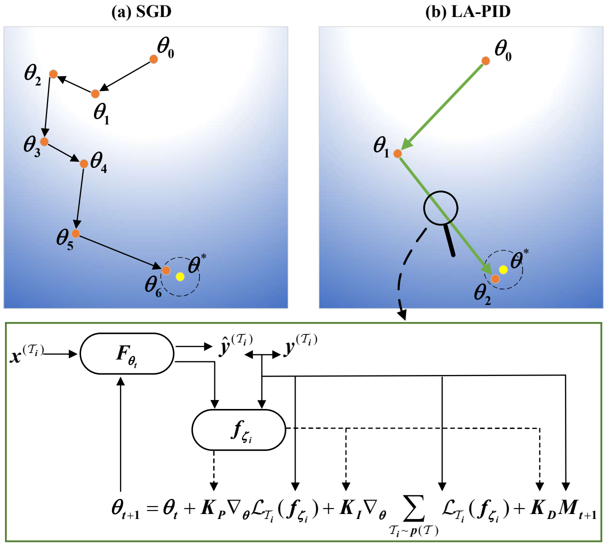
(a) SGD
(b) LA-PID
θ_0
θ_1
θ_2
θ_3
θ_4
θ_5
θ_6
θ^*
θ_0
θ_1
θ_2
θ^*
$x^{(\mathcal{T}_i)}$
F_{θ_t}
$\hat{y}^{(\mathcal{T}_i)}$
$y^{(\mathcal{T}_i)}$
f_{ζ_i}
$\theta_{t+1} = \theta_t + K_P \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_{\zeta_i}) + K_I \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\zeta_i}) + K_D M_{t+1}$

Figure 1: Overview the proposed optimization algorithm for few-shot classification tasks. (a) Traditional gradient-based optimizer (e.g., SGD) updates parameters θ: multiple steps needed to reach optimal values θ^*. (b) LA-PID achieves optimal parameter values with just two gradient updates for specific task $\mathcal{T}_i$, the green box shows the proposed gradient update rule.

Meta-learning has emerged as an effective approach for the implementation of few-shot learning (Vettoruzzo et al., 2024). This is attributable to its reinforcement of a model's capacity for heightened generalization and adaptability, consequently enabling superior performance in tasks involving limited sample sizes. Numerous studies in the field of meta-learning have been structured around a dual-level paradigm: the deliberate learning of a meta-level model for cross-task performance (Qin et al., 2023; Lin et al., 2023) and the swift acquisition of knowledge by a base-level model specific to each task(Li et al., 2023; Richards et al., 2021; Li et al., 2021; LI et al., 2023). Nevertheless, these studies usually involve the fine-tuning of various hyperparameters, including learning rate and task sampling strategies, to achieve optimal performance. This can necessitate a plethora of experiments and adjustments, thereby imposing an extra

[*] Equal contribution [1] School of Automation, Southeast University, Nanjing, China. [2] Nanjing Center for Applied Mathematics, Nanjing, China. [3] Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing, China. [4] Southeast University Shenzhen Research Institute, Shenzhen, China. Correspondence to: Xinde Li <xindeli@seu.edu.cn>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

burden on researchers. Consequently, exploring a more adaptable method for parameter optimization is a topic meriting serious consideration.

The adoption of gradient (or optimization) within a meta-learning framework has emerged as a prominent research trend, driven by its notable flexibility and efficiency. Model-agnostic meta-learning (MAML) has demonstrated remarkable performance in few-shot learning (Finn et al., 2017). Gradient descent algorithm holds a central role in deep learning, as it steers model parameters towards convergence to either the local or global minimum of the loss function, thus leading to the attainment of optimal model performance. Following this trend, numerous recent studies have sought to modify the meta-network structure to optimize the learning algorithm for quicker adaptation with limited examples (An et al., 2023; Wang et al., 2023b). However, these researches are mainly focused on learning a better initialization to achieve better generalization. Few researches have studied better optimizer for achieving rapid adaptation to new tasks, and some researches primarily focus on variants of gradient descent's weight updating step methods (Adagrad, Adam, RMS, and so on) (Jadon & Jadon, 2020). In the case of non-convex optimization, the convergence of these algorithms usually can not be guaranteed.

The application of a PID-based optimization algorithm for updating network weights has been shown to outperform other optimization methods (Wang et al., 2020a; Weng et al., 2022; Ali & Li, 2020; Dunkin et al., 2024). These studies have motivated us to seek an efficient optimization algorithm in a new domain. The proportional-integral-derivative (PID) control law, which has been widely used in various industrial and engineering applications (Wang et al., 1995; Li et al., 2020; Deng et al., 2023), has demonstrated exceptional performance in optimizing control system errors. However, the manual adjustment of PID controller parameters remains a challenge. Therefore, there is a pressing need to explore PID optimizers with parameters that can be adaptively tuned for specific tasks. Moreover, in most research, the convergence of the PID optimizer is solely demonstrated through experimental validation. The theoretical proof of convergence serves as a crucial metric for assessing the effectiveness of algorithms and also serves as the inspiration for proposing new algorithms.

Regarding these outstanding issues, we propose a novel PID-based optimization algorithm in MAML framework for few-shot classification tasks. Layer-adaptive PID optimizer (LA-PID) that dynamically updates weight parameters for each step across different network layers, allowing fast adaptability in each inner-loop gradient descent for the specific task, the overview framework is shown in Fig. 1. The main contributions of this paper can be summarized as follows:

1. LA-PID considers both the historical accumulation and future prediction of gradients, thus developing a new gradient based parameter update strategy, and it has stronger adaptability by adaptively adjusting the three hyperparameters of the PID optimizer, which is expected to expand the potential application scenarios of few-shot learning for various downstream tasks;

2. With the help of rigorous mathematical derivation, LA-PID has achieved a comprehensive analysis of the convergence of the optimizer by utilizing the dynamic characteristics of classical control systems, which provides a beneficial opportunity for the collaborative development of control theory and deep learning;

3. The comprehensive evaluation conducted on four benchmark datasets has confirmed LA-PID's state-of-the-art performance in few-shot classification and cross-domain tasks. This signifies that subsequent MAML-based approaches can draw inspiration from a broader spectrum of related disciplines, thereby offering a more diverse and promising research trajectory for future endeavors.

2. Preliminaries

To facilitate understanding of the proposed method, this section will provide a brief overview of related works in the field of few-shot learning and elucidate relevant optimization algorithms.

2.1. Related works

Few-shot learning, inspired by human-like reasoning and analytical skills, has become particularly prominent in edge computing scenarios. Initially, (Wang et al., 2020b) provided a comprehensive definition of few-shot learning in terms of machine learning experience E, task T, and performance P. As a seminal work in the field, (Finn et al., 2017) introduced the model-agnostic meta-learning (MAML) algorithm, where the model is trained by a meta-learner and can be adapted to new tasks with just a few updates. Subsequent research has seen a surge of interest in MAML-based algorithms. MAML++ (Antoniou et al., 2018) is a typical enhanced iteration of MAML that has made comprehensive improvements to address issues inherent in MAML, such as instability during training, challenging hyperparameter searches, and computational expenses. (Raghu et al., 2019) conducted an in-depth investigation into the efficacy of MAML and introduced a simplified variant named ANIL, which nearly eliminated the inner loop for all while demonstrating computational improvements over the original MAML. In recent years, numerous researches have highlighted several key challenges in training MAML and its variants (Ye & Chao, 2021). To further enhance the generalization performance of MAML, numerous investigations have been conducted on hybrid methodologies (Wang et al., 2023c; Jia et al., 2024). Nevertheless, this comes at the expense of the applicability and flexibility of MAML. The generalization performance sharply declines, especially when encountering significant distributional shifts between train and test tasks.

Despite the substantial focus on MAML-based algorithms, the optimization of these methods, particularly in refining weight-update rules, has received relatively less attention. Many recent MAML-based approaches employ simplistic inner-loop update rules without incorporating regularization, neglecting potential benefits in preventing overfitting during swift adaptation to tasks with limited samples. Few researches aimed to enhance generality by incorporating evolving gradient within the inner loop (Chen et al., 2023) and using regularization mechanism to regularize the gradients (Wang et al., 2023a). These methods, however, lack adaptability in inner-loop optimization, as their meta-learned learning rates or regularization terms do not adjust according to the specifics of each task.

Therefore, in this paper we propose a layer-adaptive PID (LA-PID) optimizer by taking all information of the gradient into consideration like PID controller. Notably, a generated network is employed to adaptively tune the three hyper-parameters of the PID optimizer for specific tasks. The convergence of the LA-PID optimizer is systematically analyzed by leveraging the dynamic characteristics of classical control system. Furthermore, we mathematically derive the initial conditions that enable the LA-PID optimizer to achieve optimal convergence performance for deep neural networks. This novel design enables LA-PID to achieve superior recognition accuracy in fewer training epochs, both in few-shot classification and cross-domain tasks.

2.2. Controller & Optimizer

To facilitate subsequent comprehension, in this section we briefly introduce the classical PID controller in the feedback control system and gradient-based optimizers in neural network.

2.2.1. PID Controller

PID controller integrate proportional control (P), integral control (I), and derivative control (D) to holistically consider error magnitude, integral value, and rate of change, respectively (Li et al., 2016). P primarily ensures a prompt response, I is applied to rectify static errors, and D focuses on mitigating overshoot and suppressing oscillations. The controller continuously assesses the error $e(t)$ at every time step t, making real-time adjustments within the feedback control system to refine performance. The classical form of PID controller can be expressed as follows.

$$u(t) = K_P e(t) + K_I \int_0^t e(t)dt + K_D \frac{d}{dt} e(t) \qquad (1)$$

where error $e(t)$ is the difference between the desired output and actual output. K_P, K_I and K_D are the controller gain coefficients of the P, I and D terms, respectively. Only through appropriately tuning the three parameters can the advantages of the controller be fully exploited, yielding a optimal control performance.

2.2.2. Gradient-based optimizers

Among gradient-based optimizers, stochastic gradient descent (SGD) optimizer has gained widespread application in updating neural network parameters due to its simplicity and efficiency. Its objective is to iteratively minimize the loss function $\mathcal{L}_t$ by adjusting the model parameters θ_t in the direction that decreases the gradient of the loss. The latest weight parameters θ_{t+1} can be obtained using the SGD update rule:

$$\theta_{t+1} = \theta_t - r \partial \mathcal{L}_t / \partial \theta_t \qquad (2)$$

where r is the learning rate in deep neural network training, θ_t is the weight parameters at iteration t, i.e., $\theta_t = \{w_{ab}, w_{bc}, w_{cd}\}$, where a, b, c and d represent different neural network layers. $\partial \mathcal{L}_t / \partial \theta_t$ is the gradient of neural network.

Remark 2.1. Comparing the PID controller (1) with SGD optimizer (2), the gradient $\partial \mathcal{L}_t / \partial \theta_t$ is similar with $e(t)$. Then the SGD optimizer is a kind of P controller with the controller coefficient $K_P = r$.

SGD-Momentum(SGD-M) is one of the successful variations of SGD. A momentum term accumulates history gradients, is introduced in gradient updates. The SGD-M optimizer ensures a more stable exploration of the optimization space, thereby expediting the training process and enhancing model performance. Its design methodology inspired our approach, and we provide a detailed theoretical derivation in Appendix A for reference. The update rule of network parameters can be written as follows.

$$\theta_{t+1} = \theta_t - r \partial L_t / \partial \theta_t - r \sum_{m=0}^{t-1} \partial L_m / \partial \theta_m \alpha^{t-m}. \qquad (3)$$

where $\alpha \in [0, 1]$ is the momentum coefficient, especially when $\alpha = 0$, SGD-M becomes mini-bach GD (Li et al., 2014).

Remark 2.2. Comparing the PID controller (1) with SGD-M optimizer (3), it is worth noting that the present gradient $\partial L_t / \partial \theta_t$ and the accumulation of history gradients $\sum_{m=0}^{t-1} \partial L_m / \partial \theta_i \alpha^{t-m}$ are correspond to the proportional item and integral item of PID controller, respectively.

From the aforementioned related works, it is evident that the gradient optimizer plays a crucial role in the training process of network weights. The update rules for network parameters are primarily designed based on gradient information, sharing a similarity with the design of PID controllers that rely on system error. Both approaches exhibit a common thread in their design philosophy. These provide us with insights for designing a novel optimizer in the next section.

3. Proposed Method

Building upon the preceding analyses of PID controllers and gradient-based optimizers, this section presents the comprehensive design details of the LA-PID optimizer. The proposed overview framework is shown in Fig. 1. The core idea of this method is expedite the parameter updates of the deep neural network. A PID-like optimizer is implemented within the model architecture, which takes into account the current, past, and future information of the gradient (i.e., the partial derivative of the loss function $\mathcal{L}$ with respect to weight parameters θ_t), aiming to achieve adaptively rapid convergence of weight parameters during the training process.

3.1. LA-PID Optimizer

Inspired from PID controller (1), we integrate a differential item ϕ_{t+1} into SGD-M optimizer, aiming to achieve an excellent performance akin to PID control system, including reduced overshoot and minimized steady-state error in system state responses. For a new sampled task $\mathcal{T}_i$, the proposed LA-PID optimizer updates parameter θ_t^i according to the following rules. For convenience, the superscript i is abbreviated hereafter.

$$\begin{cases} M_{t+1} = \alpha M_t - r\partial L_t/\partial\theta_t \\ \phi_{t+1} = \alpha\phi_t + (1-\alpha)(\partial L_t/\partial\theta_t - \partial L_{t-1}/\partial\theta_{t-1}) \\ \theta_{t+1} = \theta_t + M_{t+1} + K_D\phi_{t+1}. \end{cases} \tag{4}$$

where ϕ_{t+1} is a term that predicts future gradients, K_D is a hyperparameter to be designed.

From (1), (2), (3) and (4), the latest network parameter θ_{t+1}^i can be updated using the PID optimization rule:

$$\begin{aligned} \theta_{t+1} = \theta_t^i - K_P\partial L_t/\partial\theta_t - K_I\sum_{m=0}^{t-1}\partial L_m/\partial\theta_m\alpha^{t-m} \\ - K_D\sum_{m=0}^{t-1}(\alpha^{t-m} - \alpha^{t-m-1})(\partial L_t/\partial\theta_t^i - \partial L_{t-1}/\theta_{t-1}^i) \end{aligned} \tag{5}$$

where the three hyperparameters K_P, K_I and K_D can be generated by generated network f_{ζ_i}. Essentially, these hyperparameters are functions parameterized by the learning rate r. For convenience, the superscript i is abbreviated hereafter.

To address the challenge of tuning parameters in the classical PID controller, we employ a generated network f_{ζ_i} to adaptively tune the three hyperparameters of the LA-PID optimizer for specific tasks. We assert that distinct network layers capture features at varying levels of granularity, necessitating individualized learning rates and hyperparameters for each layer.

Referring to the initialization of training weight parameter θ in MAML, LA-PID is implemented in the inner-loop updating. To enable the model equipped with the considerably rapid learning ability that adapt to new scenario, we believe that different layer of the base backbone should be endowed with specific learning rate so that unleash the potential of the entire network. Referring to the design of hyperparameter generator in (Baik et al., 2023). The generated network f_{ζ_i} has a two-layer MLP neural network with a ReLU activation function connecting the layers. Furthermore, we consider the prospective effective information of the network is involved in the layer parameters: the mean, variance and gradients which are taken of the input of f_{ζ_i}. The neural network f_{ζ_i} with network parameters ζ_i can be written as follows:

$$[K_P, K_I, K_D] = f_\zeta(\bar{\theta}_t, \hat{\theta}_t, G_t; \zeta) \tag{6}$$

where $\bar{\theta}_t = \{\bar{\theta}_t^k\}^{k=1,\cdots,N}$, $\hat{\theta}_t = \{\hat{\theta}_t^k\}^{k=1,\cdots,N}$, $G_t = \{G_t^k\}^{k=1,\cdots,N}$, N is the number layers of the network backbone, $\bar{\theta}_t$, $\hat{\theta}_t$ and G_t are the mean, variance and the gradient at the k-th layer of backbone network parameters.

Besides, based on different specific tasks $\mathcal{T}_i'$ and the optimized base-learner F_{θ_t}, the generated network parameter ζ_i update rule can be designed as

$$\zeta_i \leftarrow \zeta_i - \beta\sum_{\mathcal{T}_i'}\partial L_{\mathcal{T}_i'}/\partial\theta_t \tag{7}$$

The base-learner F_{θ_t} is updated the network parameter by using the LA-PID optimization rule (5).

The pseudo-codes for the training procedure of LA-PID hyperparameters is summarized in Algorithm 1.

3.2. Initialization of Hyperparameters K_P, K_I, K_D

In Theorem 3.1, we conclude the optimal theoretical hyperparameters initialization range for LA-PID optimizer. And, the system dynamics can be characterized by these hyperparameters.

Theorem 3.1. *For some given positive real number α, r, if there exist functions K_P, K_I and K_D that depend on the independent variable r, and they satisfy $0 < \frac{K_P+1}{2\sqrt{K_I K_D}} < 1$, such that the LA-PID optimizer is a second-order control system, with the transfer function*

$$W_B(s) = \frac{1}{K_D}\cdot\frac{K_I/K_D}{s^2 + \frac{K_P+1}{K_D}s + \frac{K_I}{K_D}} \tag{8}$$

And, the dynamics is determined by the damping coefficient $\xi = \frac{K_P+1}{2\sqrt{K_I K_D}}$, *oscillation frequency* $\omega_n = \sqrt{\frac{K_I}{K_D}}$, *overshoot* $\sigma\% = \exp\left(-\frac{\pi}{\sqrt{\frac{4K_I K_D}{(K_P+1)^2}-1}}\right) \times 100\%$, *setting time* $t_s \approx \frac{6K_D}{K_P+1}$, *oscillation period* $T = \frac{2\pi}{\sqrt{\omega_n\sqrt{1-\xi^2}}}$.

Algorithm 1 Layer-Adaptive PID (LA-PID) Learning

```
Require: Task distribution $p(\mathcal{T}_i)$, outer-loop optimization rate $\beta$
Randomly initialize $\zeta_i$.
while not converged do
  Sample a new tasks $\mathcal{T}_i \sim p(\mathcal{T}_i)$
  for each sampled task $\mathcal{T}_i$ do
    Random initialize $\theta_0$
    for inner-loop optimization step $t = 0$ to $STEP-1$ do
      Compute the loss function $\partial L_{\mathcal{T}_i}/\partial\theta_t$
      Compute $f_{\zeta_i}$ learning state $[\bar{\theta}_t, \hat{\theta}_t, G_t]$
      Compute PID hyperparameters $(K_P, K_I, K_D)$
      Compute the latest weight parameter with LA-PID optimization rule (5)
    end for
    Compute $\partial L'_{\mathcal{T}_i}/\partial\theta'_t$ by evaluating $\partial L_{\mathcal{T}_i}/\partial\theta_t$ w.r.t. a query set from $\mathcal{T}'_i$
  end for
  Perform gradient descent to update parameters: (7)
end while
```

The details of theoretical analyses are provided in Appendix B. Generally, increasing the derivate term can speed up neural network training, but excessive values may make the system fragile. Referring to the Ziegler-Nichols rule (Ziegler & Nichols, 1942), the optimal derivate-action coefficient set $K_D = \frac{1}{3}T$. Simplifying the decay term $\alpha = 1$ in momentum, then we can get $K_P = r$ and $K_I = r$ from (3). Combined with Theorem 3.1, the hyperparameter K_D can be derived in a specific solution form

$$K_D = \frac{1}{4}r + (1 + \frac{16}{9}\pi^2)\frac{1}{r} + \frac{1}{2}. \tag{9}$$

Remark 3.2. The closed-loop transfer function (8) provides an approximation of network parameter model from the perspective of a control system. The coverage performance of the designed LA-PID controller is more excellent compared to both the traditional PID controller and the original SGD and SGD-M optimizer.

Remark 3.3. The stability of deep neural network model (4) is not only affected by network architecture, but also the setup of optimization strategy. LA-PID optimizer is initialized with the ideal settings of hyperparameters K_P, K_I and K_D, and then adaptively tunes these three hyperparameters of optimization rule for different application scenarios.

4. Experiments

In this section, we compare our proposed LA-PID optimizer with recent state-of-the-art methods on several benchmark datasets for few-shot image classification. The results demonstrate the effectiveness and superiority of our proposed LA-PID algorithm in the realm of few-shot learning. Even when tested in cross-domain scenarios with significantly different distributions between training and testing tasks, the neural network based on LA-PID can achieve substantial classification accuracy with just fewer training epochs. This indicates that the LA-PID-based gradient update rule effectively enhances the generalization capabilities of few-shot learning models.

4.1. Datasets

In our experiment, we preprocess the four benchmark datasets as follows: **mini-ImageNet** (Vinyals et al., 2016) consists of 100 classes with 60,000 RGB images of size 84 × 84. The dataset is partitioned into three non-overlapping subsets: 64 classes for training, 16 for validation, and 20 for testing. **tiered-ImageNet** (Ren et al., 2018) is composed of 608 classes with 1281 samples of 84 × 84 RGB images. To maintain dissimilarity between the training and testing sets, the dataset is divided into 20 training (351 classes), 6 validation (97 classes), and 8 test (160 classes) categories. **CIFAR-FS** (Bertinetto et al., 2018) includes a total of 100 classes, each with 600 images sized 32 × 32 pixels. The dataset is typically split into training, validation, and test meta-sets, with 64, 16, and 20 classes, respectively. **FC100** (Oreshkin et al., 2018) is composed of 100 classes with 60,000 images. FC-100 contains a total of 20 super classes (60 classes), of which 12/4/4 super classes for training/validation/test set.

Moreover, to evaluate the rapid learning and generalization capabilities for few-shot models, a cross-domain scenario is introduced in (Chen et al., 2019) to simulate real-world few-shot tasks environment, in which the distribution between training datasets and the test datasets are substantially different. Specially, LA-PID was trained on the mini-ImageNet but tested on CUB-200-2011(denoted as **CUB**) (Wah et al., 2011), which is a fine-grained dataset involving with 200 birds classes.

4.2. Implementation details

For our network architecture, we employ a 4-layer Convolutional Neural Network (4-CONV) and ResNet12 as the feature extraction backbone, adhering to the same experimental settings as in (Rusu et al., 2018; Sung et al., 2018). The model is trained for 30 epochs and each epoch with 500 iterations, we set the batch size of 2 and 4 for 5-shot and 1-shot, respectively. For N-way K-shot classification tasks, N categories are randomly selected, each categories with

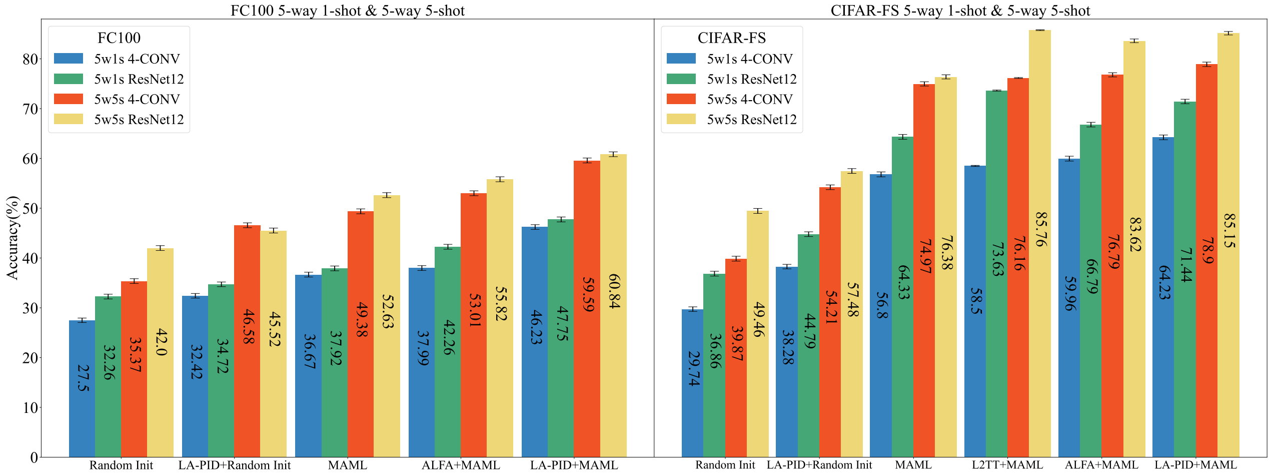

Figure 2: Test accuracy on 5-way classification for FC100 and CIFAR-FS with the backbone of 4-CONV and ResNet12.

Table 1: Test accuracy on 5-way classification for mini-ImageNet and tiered-ImageNet.

	Backbone	mini-ImageNet		tiered-ImageNet	
		1-shot	5-shot	1-shot	5-shot
Random Init	4-CONV	24.85±0.43%	31.09±0.46%	26.55±0.44%	33.82±0.47%
LA-PID +Random Init		**29.16±0.45%**	**38.43±0.48%**	**30.28±0.45%**	**47.73±0.49%**
MAML	4-CONV	48.70±1.75%	63.11±0.91%	49.06±0.50%	67.48±0.47%
LA-PID +MAML		**57.48±0.49%**	**72.02±0.44%**	**55.11±0.49%**	**71.99±0.44%**
ALFA+MAML	4-CONV	50.58±0.51%	69.12±0.47%	53.16±0.49%	70.54±0.46%
L2TT+MAML		47.70±0.10%	64.75±0.09%	-	-
MAML+Finetune		-	64.87±0.40%	-	-
Random Init	ResNet12	31.23±0.46%	41.60±0.49%	33.46±0.47%	44.54±0.50%
LA-PID +Random Init		**44.32±0.47%**	**51.46±0.49%**	**39.02±0.48%**	**55.05±0.49%**
MAML	ResNet12	58.37±0.49%	69.76±0.46%	58.58±0.49%	71.24±0.43%
LA-PID +MAML		**63.29±0.48%**	**79.18±0.43%**	**64.77±0.47%**	**82.59±0.37%**
ALFA+MAML	ResNet12	59.74±0.49%	77.96±0.41%	64.62±0.49%	82.48±0.38%
L2TT+MAML		60.82±0.11%	78.16±0.08%	-	-
MAML+Finetune		-	73.13±0.40%	-	-

K labeled training samples, 15 query samples are sampled from others samples for per category during each iteration. Furthermore, we implement a cosine annealing learning rate drop strategy for the meta-optimizer, starting with an initial learning rate of 0.01 and reducing it to a minimum of 5×10^{-4} in the outer-loop, The LA-PID optimizer is utilized in the inner-loop to update the learnable parameters.

4.3. Experimental results

4.3.1. Few-shot classification

Table 1 and Fig. 2 present the results of testing LA-PID on various datasets, including mini-ImageNet, tiered-ImageNet, FC100, and CIFAR-FS, under different initialization settings, such as Random Init, MAML, and ALFA. The experiment also includes comparisons with other existing state-of-the-art meta-learning algorithms (ALFA (Baik et al., 2023), L2TT (Chen et al., 2019)), which used to prove the effectiveness and superiority of the LA-PID when processing few-shot learning tasks. Moreover, due to the differences in both class-intra hierarchy and image resolution (84 × 84 and 32 × 32) between mini-ImageNet and tiered-ImageNet, as well as FC-100 and CIFAR-FS, LA-PID enhances performance significantly. This demonstrates the universality and generalization of the proposed inner-loop gradient update rule. Specially, LA-PID is trained for fewer epochs than ALFA and MAML, indicating that the proposed method possesses rapid learning abilities to adapt new domain tasks, proving that robust gradient updating strategy is critical in ensuring model stability, which can be clearly see in Fig 3.

To further evaluate the universality of the method, we test LA-PID on mini-ImageNet by the 20-way 5-shot task, the results are exhibited in the Table 2, demonstrating that LA-PID optimizer remains superior and effective even in unknown

Table 2: 20-way classification 4-conv mini-ImageNet.

Model	1-shot	5-shot
MAML	15.21±0.36%	18.23±0.39%
ALFA+MAML	22.03±0.41%	35.33±0.48%
LA-PID +MAML	**34.04±0.47%**	**43.66±0.43%**

experiment settings. This finding reignites the effectiveness of the PID gradient update mechanism in the backpropagation of neural networks.

To better visualize the classification performance of the proposed optimization algorithm (LA-PID+MAML), we randomly select 5 categories from the mini-ImageNet test set and employ 4-CONV as backbone network. Subsequently, T-SNE is utilized to perform feature dimensionality reduction on the output results for visualization purposes. The distribution in the feature space is depicted in Fig. 4. Despite the classification accuracy being 72.02% using the proposed optimization algorithm (LA-PID+MAML), the distinction between different clusters is clear.

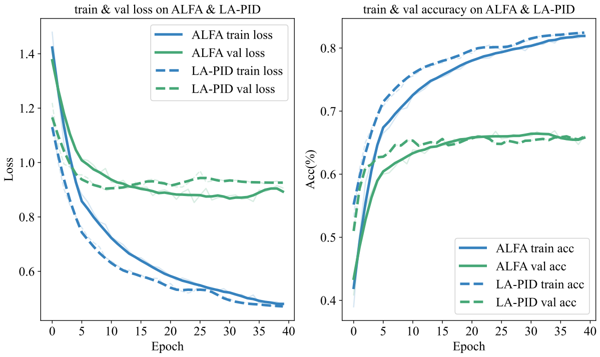

Figure 3: The training and validation curves of LA-PID and ALFA on mini-ImageNet reveal that LA-PID achieves convergence in significantly fewer training iterations.

Upon comparing the training and validation curves of LA-PID and ALFA on mini-ImageNet, it becomes clear that LA-PID converges more rapidly and attains a higher validation accuracy than ALFA. This underscores the efficiency and superiority of the LA-PID method, which can be attributed to its innovative gradient computation and parameter updating mechanism.

4.3.2. Cross-domain Few-shot classification

To further validate the generalization and rapid adaptation capabilities of LA-PID's gradient update mechanism, we conduct experiments on cross-scenario tasks where there is a significant distribution discrepancy between the training and the testing dataset. Specifically, the model was trained on mini-ImageNet and tested on CUB, a fine-grained dataset focusing on birds with 11,788 images and 200 bird subclasses. The test accuracy for 5-way 5-shot for cross-domain classification with the 4-CONV backbone is exhibited in the Table 6, revealing that the classification accuracy decreases when the training and testing tasks are inconsistent. Nonetheless, LA-PID's performance surpasses current state-of-the-art methods, highlighting the importance of the adaptive gradient update rule in inner-loop for cross-domain tasks. Surprisingly, even when using ResNet12 as the backbone for feature extraction, the cross-domain classification accuracy exceeds that of ResNet18, further demonstrating the superiority of the LA-PID method.

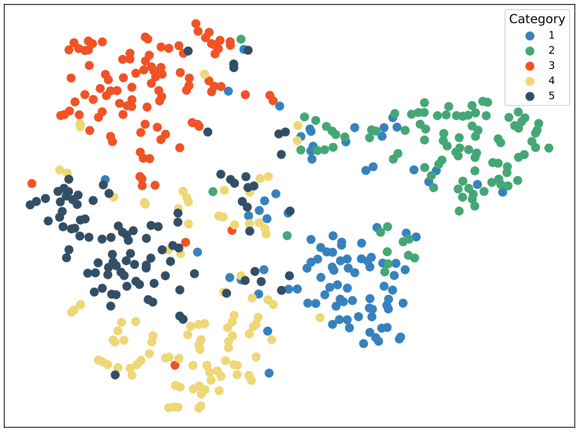

Figure 4: T-SNE Visualization of mini-lmageNet Dataset.

4.4. Ablation studies

In this section, we conduct ablation studies to gain a deeper understanding of the effectiveness of the proposed LA-PID optimizer. These studies are performed under experimental settings designed for 5-way 1-shot and 5-shot classification tasks on the mini-ImageNet dataset.

4.4.1. Ablation study on Hyperparameters

In order to evaluate the impact of adaptively generated hyperparameters on classification accuracy, we fix hyperparameter $K_P = 1$ and vary only the values of K_I and K_D. The test accuracy experiments for 5-way 5-shot and 1-shot scenarios on the mini-ImageNet, tiered-ImageNet, FC-100 and CIFAR-FS datasets are shown in Table 3. The results, obtained using both the 4-CONV and ResNet12 backbones, demonstrate a substantial performance improvement when generating these hyperparameters simultaneously.

4.4.2. innner-loop step

To further dissect the efficiency of LA-PID's learning process step by step, we conducted 5-way 5-shot experiments on mini-ImageNet using a 4-CONV network backbone. The

Table 3: Ablation study on K_I, K_D for different datasets

	K_I	K_D	mini-ImageNet		tiered-ImageNet		FC100		CIFAR-FS	
			5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
4-CONV	✓		50.87±0.50%	68.34±0.46%	48.05±0.49%	68.83±0.46%	37.78±0.48%	53.36±0.49%	58.67±0.49%	75.09±0.43%
4-CONV		✓	50.58±0.49%	68.52±0.46%	48.67±0.48%	69.04±0.46%	38.46±0.48%	53.56±0.49%	58.50±0.49%	74.72±0.43%
4-CONV	✓	✓	**57.48±0.49%**	**72.02±0.44%**	**55.11±0.49%**	**71.99±0.44%**	**46.23±0.47%**	**59.59±0.49%**	**64.23±0.47%**	**78.90±0.45%**
ResNet12	✓		58.17±0.49%	75.87±0.42%	61.75±0.48%	79.80±0.40%	41.47±0.49%	55.40±0.49%	68.16±0.46%	57.73±0.49%
ResNet12		✓	58.40±0.49%	73.96±0.43%	59.07±0.49%	79.67±0.40%	39.64±0.48%	55.21±0.49%	66.61±0.47%	57.72±0.49%
ResNet12	✓	✓	**63.29±0.48%**	**79.18±0.43%**	**64.77±0.47%**	**82.59±0.37%**	**47.75±0.49%**	**60.84±0.48%**	**71.44±0.45%**	**85.15±0.35%**

Table 4: The number of inner-loop steps for fast adaptation.

MAML	**LA-PID+MAML**					
Step5	Step1	Step2	Step3	Step4	Step5	Step6
63.11±0.91%	70.21±0.45%	70.65±0.45	70.94±0.45%	71.19±0.45%	72.01±0.44%	72.02±0.44%
63.11±0.91%	66.37±0.47%	69.09±0.46	70.62±0.45%	71.68±0.45%	71.91±0.44%	71.94±0.44%

Table 5: Ablation study on f_ζ.

Input	5-way 5-shot
weight only	63.79±0.48%
gradient only	64.10±0.47%
weight + gradient (**LA-PID**)	**72.02±0.44%**

Table 6: Test accuracy on 5-way 5-shot cross-domain classification.

	Backbone	**mini-ImageNet→CUB**
MAML	4-CONV	52.70±0.32%
LA-PID +MAML		**59.73±0.49%**
ALFA+MAML		58.35±0.25%
MAML	ResNet12	53.83±0.32%
LA-PID +MAML		**65.93±0.47%**
ALFA+MAML		61.22±0.22%
Baseline	ResNet18	65.57±0.70%
Baseline++		62.04±0.76%
MAML		51.34±0.72%

results in Table 4 reveal that with a single-step update in the inner loop, the test set accuracy surpasses that of MAML with five inner loop step updates. Moreover, comparing the experimental outcomes of two instances of LA-PID, it is evident that even if the initial update in the second experiment fails to capture optimal features, leading to reduced recognition accuracy, subsequent updates can swiftly adjust and attain high performance. Typically, parameter convergence is achieved by the fourth inner loop update, illustrating LA-PID's rapid learning capability and its stable gradient updating mechanism. The final results for both experiments exhibit an absolute error within 0.1%, indicating the robustness and stability of the proposed method. Even with random disturbances in the hyperparameter-generating network, network weight updating converges to the same accuracy level under the LA-PID framework.

4.4.3. Generated network

To assess the influence of the input information on the hyperparameter generation network f_{ζ_i}, a 5-way 5-shot few-shot classification experiment is conducted on the mini-ImageNet using a 4-CONV backbone. The results, reported in Table 5, demonstrate the effects of altering the input information for f_{ζ_i} for the final classification performance on the test set. This experiment reveals that superior recognition performance is achieved when the mean, variance, and gradients of the θ are used as inputs to the generation network f_{ζ_i}. This finding underscores the importance of parameter information and their gradients in guiding the learning direction of the network model.

It is noteworthy that the information contained in individual parameters and gradients is relatively localized. Hence, employing them together provides a more comprehensive representation of the overall learning direction of the network. This aligns with our understanding that comprehending the learning direction of the model is essential. Simultaneously, understanding the current learning state of the network is equally crucial. The combination of both aspects enables a more comprehensive determination of the overall optimization direction for the model.

4.4.4. Memory usage

Despite the significant performance enhancement achieved by LA-PID in few-shot image classification and cross-domain tasks, it is important to assess whether LA-PID incurs a larger memory footprint. To this end, we conducted tests on the mini-ImageNet dataset using both 4-CONV and ResNet12 as the network backbone. We modified LA-PID to calculate only the current and previous errors, omitting the historical cumulative error, and refer to this variant as ***LA-PID++***. The experimental results, presented in the table 7, indicate that the memory usage of the LA-PID is nearly on par with that of the baseline. Moreover, the enhanced LA-PID++ exhibits a memory footprint that is virtually identical to the baseline, confirming LA-PID's superiority and effectiveness in terms of memory efficiency.

Table 7: The memory footprint compared to the baseline

Dataset	Method	Exp-setting	Backbone	Batchsize	Inputsize	Learnableparams	GPUmemory(M)	Acc(%)
mini-ImageNet	Baseline	5w1s	4-CONV	4	84×84×3	70145	2318	48.70
mini-ImageNet	LA-PID	5w1s	4-CONV	4	84×84×3	70205	2944	57.48
mini-ImageNet	**LA-PID++**	5w1s	4-CONV	4	84×84×3	70205	2451	55.83
mini-ImageNet	Baseline	5w1s	ResNet12	4	84×84×3	7999473	9647	58.37
mini-ImageNet	LA-PID	5w1s	ResNet12	4	84×84×3	7999581	11966	63.29
mini-ImageNet	**LA-PID++**	5w1s	ResNet12	4	84×84×3	7999581	9705	62.04
mini-ImageNet	Baseline	5w5s	4-CONV	2	84×84×3	70145	1683	63.11
mini-ImageNet	LA-PID	5w5s	4-CONV	2	84×84×3	70205	2094	72.02
mini-ImageNet	**LA-PID++**	5w5s	4-CONV	2	84×84×3	70205	1738	71.57
mini-ImageNet	Baseline	5w5s	ResNet12	2	84×84×3	7999473	6071	69.76
mini-ImageNet	LA-PID	5w5s	ResNet12	2	84×84×3	7999581	7482	79.18
mini-ImageNet	**LA-PID++**	5w5s	ResNet12	2	84×84×3	7999581	6182	77.85

4.5. Limitations

We visualize the parameters generated by the generation network f_ζ. For a clearer view of specific parameter changes, K_P is fixed to 1. The generated values for K_I and K_D are shown in Fig. 5. It can be observed that only a few layers have non-zero values for the generated parameters. This indicates that the generation algorithm does not fully utilize information from the entire network but rather only had an impact on certain layers. Future work would focus on optimizing PID parameter generation to enhance the efficiency, such as considering activation-based generation rather than applying it to all layers.

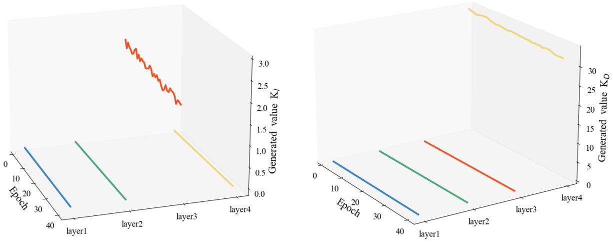

Figure 5: Visualized the hyperparameters generated by the LA-PID. Not every layer of the model needs to generate PID parameters, as evidenced by the visualization of non-zero PID parameters.

5. Conclusion

In this paper we explore the connection between gradient-based parameter optimization strategy and PID control theory, subsequently introduce a LA-PID optimizer within the MAML framework. Our approach endows the network model with the advantages of the classical PID control system, including stability, accuracy, and rapidity. Notably, the hyperparameters of proposed optimizer are adaptively tuned through inner-loop optimization for specific tasks. Experiment results demonstrate that our proposed algorithm achieves a state-of-the-art performance across benchmark datasets. We assert that setting a specific optimal weight-update rule for different recognition tasks is as crucial as designing complex network backbones. Nevertheless, the generation algorithm does not fully leverage all available network resources, this also serves as a direction for our future work, aiming to leverage the latent information across the entire network.

Acknowledgements

We extend our heartfelt gratitude to Chuanfei Hu for offering invaluable spiritual support throughout the work. Furthermore, we are deeply appreciative of the reviewers for their constructive feedback, which has been instrumental in enhancing the quality of our paper.

This work was co-supported by the National Natural Science Foundation of China (Nos. 62233003 and 62073072), the Key Projects of Key R&D Program of Jiangsu Province, China (Nos. BE2020006 and BE2020006-1), and the Shenzhen Science and Technology Program, China (Nos. JCYJ20210324132202005 and JCYJ20220818101206014).

Impact Statement

This paper aims to streamline the training process and bolster the precision of few-shot learning cross-domains tasks. The conventional paradigm of supervised learning, which relies heavily on a substantial corpus of labeled data, is not only labor-intensive and financially demanding in terms of human effort and labeling costs, but also often unfeasible due to the scarcity of labeled samples in practical applications. The quest to refine the efficiency and accuracy of learning from a modest dataset is pivotal to the ongoing evolution of machine learning. It is our aspiration that the contributions of this work will catalyze further advancements in the field.

References

Aggarwal, P., Deshpande, A., and Narasimhan, K. R. Semsup-xc: semantic supervision for zero and few-shot extreme classification. In *International Conference on Machine Learning*, pp. 228–247. PMLR, 2023.

Ali, Z. A. and Li, X. Controlling of an under-actuated quadrotor uav equipped with a manipulator. *IEEE Access*, 8:34664–34674, 2020.

An, Y., Xue, H., Zhao, X., and Wang, J. From instance to metric calibration: A unified framework for open-world few-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):9757–9773, 2023. doi: 10.1109/TPAMI.2023.3244023.

Antoniou, A., Edwards, H., and Storkey, A. How to train your MAML. *arXiv preprint arXiv:1810.09502*, 2018.

Baik, S., Choi, M., Choi, J., Kim, H., and Lee, K. M. Learning to learn task-adaptive hyperparameters for few-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Bertinetto, L., Henriques, J. F., Torr, P. H., and Vedaldi, A. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

Chen, J., Yuan, W., Chen, S., Hu, Z., and Li, P. Evo-maml: Meta-learning with evolving gradient. *Electronics*, 12 (18):3865, 2023.

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C. F., and Huang, J.-B. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

Deng, J., Liu, H., Fang, H., Shao, S., Wang, D., Hou, Y., Chen, D., and Tang, M. Mgnet: A fault diagnosis approach for multi-bearing system based on auxiliary bearing and multi-granularity information fusion. *Mechanical Systems and Signal Processing*, 193:110253, 2023.

Dunkin, F., Li, X., Hu, C., Wu, G., Li, H., Lu, X., and Zhang, Z. Like draws to like: A multi-granularity ball-intra fusion approach for fault diagnosis models to resists misleading by noisy labels. *Advanced Engineering Informatics*, 60:102425, 2024.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135. PMLR, 2017.

Jadon, S. and Jadon, A. An overview of deep learning architectures in few-shot learning domain. *arXiv preprint arXiv:2008.06365*, 2020.

Jia, J., Feng, X., and Yu, H. Few-shot classification via efficient meta-learning with hybrid optimization. *Engineering Applications of Artificial Intelligence*, 127:107296, 2024.

Li, M., Zhang, T., Chen, Y., and Smola, A. J. Efficient mini-batch training for stochastic optimization. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 661–670, 2014.

Li, X., Luo, C., Xu, Y., and Li, P. A fuzzy pid controller applied in agv control system. In *2016 International Conference on Advanced Robotics and Mechatronics (ICARM)*, pp. 555–560. IEEE, 2016.

Li, X., Li, X., and Pan, H. Multi-scale vehicle detection in high-resolution aerial images with context information. *IEEE Access*, 8:208643–208657, 2020.

Li, X., Li, X., Li, Z., Xiong, X., Khyam, M. O., and Sun, C. Robust vehicle detection in high-resolution aerial images with imbalanced data. *IEEE Transactions on Artificial Intelligence*, 2(3):238–250, 2021.

LI, X., DUNKIN, F., and DEZERT, J. Multi-source information fusion: Progress and future. *Chinese Journal of Aeronautics*, 2023.

Li, Z., Tang, H., Peng, Z., Qi, G.-J., and Tang, J. Knowledge-guided semantic transfer network for few-shot image recognition. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2023. doi: 10.1109/TNNLS.2023.3240195.

Lin, Z., Yu, S., Kuang, Z., Pathak, D., and Ramanan, D. Multimodality helps unimodality: Cross-modal few-shot learning with multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19325–19337, 2023.

Luo, X., Wu, H., Zhang, J., Gao, L., Xu, J., and Song, J. A closer look at few-shot classification again. *arXiv preprint arXiv:2301.12246*, 2023.

Oreshkin, B., Rodríguez López, P., and Lacoste, A. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems*, 31, 2018.

Qin, C., Joty, S., Li, Q., and Zhao, R. Learning to initialize: Can meta learning improve cross-task generalization in prompt tuning? *arXiv preprint arXiv:2302.08143*, 2023.

Raghu, A., Raghu, M., Bengio, S., and Vinyals, O. Rapid learning or feature reuse? towards understanding the effectiveness of MAML. *arXiv preprint arXiv:1909.09157*, 2019.

Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J. B., Larochelle, H., and Zemel, R. S. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

Richards, S. M., Azizan, N., Slotine, J.-J., and Pavone, M. Adaptive-control-oriented meta-learning for nonlinear systems. *arXiv preprint arXiv:2103.04490*, 2021.

Rusu, A. A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., and Hadsell, R. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.

Song, Y., Wang, T., Cai, P., Mondal, S. K., and Sahoo, J. P. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Computing Surveys*, 2023.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1199–1208, 2018.

Vettoruzzo, A., Bouguelia, M.-R., Vanschoren, J., Rognvaldsson, T., and Santosh, K. Advances and challenges in meta-learning: A technical review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2024. doi: 10.1109/TPAMI.2024.3357847.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. 2011.

Wang, H., Luo, Y., An, W., Sun, Q., Xu, J., and Zhang, L. PID controller-based stochastic optimization acceleration for deep neural networks. *IEEE transactions on neural networks and learning systems*, 31(12):5079–5091, 2020a.

Wang, L., Barnes, T., and Cluett, W. R. New frequency-domain design method for PID controllers. *IEE proceedings-control theory and applications*, 142(4):265–271, 1995.

Wang, L., Zhou, S., Zhang, S., Chu, X., Chang, H., and Zhu, W. Improving generalization of meta-learning with inverted regularization at inner-level. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7826–7835, 2023a.

Wang, X., Du, Y., Chen, D., Li, X., Chen, X., Fan, Y., Xie, C., Li, Y., Liu, J., and Li, H. Dual adversarial network with meta-learning for domain-generalized few-shot text classification. *Applied Soft Computing*, 146: 110697, 2023b.

Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys*, 53(3):1–34, 2020b.

Wang, Y., Yan, J., Yang, Z., Qi, Z., Wang, J., and Geng, Y. A novel hybrid meta-learning for few-shot gas-insulated switchgear insulation defect diagnosis. *Expert Systems with Applications*, 233:120956, 2023c.

Weng, B., Sun, J., Sadeghi, A., and Wang, G. Adapid: An adaptive pid optimizer for training deep neural networks. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3943–3947, 2022. doi: 10.1109/ICASSP43922.2022.9746279.

Ye, H.-J. and Chao, W.-L. How to train your maml to excel in few-shot classification. *arXiv preprint arXiv:2106.16245*, 2021.

Ziegler, J. G. and Nichols, N. B. Optimum settings for automatic controllers. *Transactions of the American society of mechanical engineers*, 64(8):759–765, 1942.

A. SGD-M Optimizer

SGD-Momentum(SGD-M) is one of the successful variations of SGD. A momentum term M_t, which accumulates history gradients, is introduced in gradient updates. The SGD-M optimizer ensures a more stable exploration of the optimization space, thereby expediting the training process and enhancing model performance. The update rule of network parameters can be written as follows.

$$\begin{cases} M_{t+1} = \alpha M_t - r\partial L_t/\partial\theta_t \\ \theta_{t+1} = \theta_t + M_{t+1} \end{cases} \tag{10}$$

where $\alpha \in [0, 1]$ is the momentum coefficient, especially when $\alpha = 0$, SGD-M becomes mini-bach GD (Li et al., 2014). Set initial condition $M_0 = 0$.

Referring to (Wang et al., 2020a), we conduct the following mathematical operations.

Dividing the first equation of (10) yields,

$$\frac{M_{t+1}}{\alpha^{t+1}} = \frac{M_t}{\alpha^t} - r\frac{\partial L_t/\partial\theta_t}{\alpha^{t+1}}. \tag{11}$$

Expanding Equation (11) from iteration time $t + 1$ to 1 yields

$$\begin{cases} \frac{M_{t+1}}{\alpha^{t+1}} - \frac{M_t}{\alpha^t} = -r\frac{\partial L_t/\partial\theta_t}{\alpha^{t+1}} \\ \frac{M_t}{\alpha^t} - \frac{M_{t-1}}{\alpha^{t-1}} = -r\frac{\partial L_{t-1}/\partial\theta_{t-1}}{\alpha^t} \\ \vdots \\ \frac{M_1}{\alpha^1} - \frac{M_0}{\alpha^0} = -r\frac{\partial L_0/\partial\theta_0}{\alpha^1}. \end{cases} \tag{12}$$

Summarizing the equations (12) yields

$$\frac{M_{t+1}}{\alpha^{t+1}} = \frac{M_0}{\alpha^0} - r\sum_{i=0}^{t}\frac{\partial L_i/\partial\theta_i}{\alpha^{i+1}} \tag{13}$$

Under the initial condition $M_0 = 0$, multiplying both sides of Equation (13) by α^{t+1} yields

$$M_{t+1} = -r\sum_{m=0}^{t}\alpha^{t-m}\partial L_m/\partial\theta_m \tag{14}$$

Extracting the present gradient $\partial L_t/\partial\theta_t$ from Equation (14), which is convenient to subsequent analysis. The finial form of M_{t+1} can be written as follows.

$$M_{t+1} = -r\partial L_t/\partial\theta_t - r\sum_{m=0}^{t-1}\partial L_m/\partial\theta_m\alpha^{t-m}. \tag{15}$$

Substituting (15) into (10), we can derive the standard SGD-M, which consists of present and history gradient information.

$$\theta_{t+1} = \theta_t - r\partial L_t/\partial\theta_t - r\sum_{m=0}^{t-1}\partial L_m/\partial\theta_m\alpha^{t-m}. \tag{16}$$

B. Proof for Theorem 3.1

Herein, we furnish an exhaustive proof for Theorem 3.1. The connection between the LA-PID optimizer and the PID controller is established using Laplace transform. The detailed theoretical proofs unfold from **Case 1** to **Case 5**, where **Case 3** is the optimal convergence result.

Proof. Firstly, the initial state of $\theta(t)$ is defined as θ_0, the optimal value of $\theta(t)$ is θ^* can be obtained after enough epochs of training.

The Laplace transform is given as $\mathfrak{L}(\theta^*) = \theta^*/s$, $\mathfrak{L}(\theta(t)) = \theta(s)$. Then, the time domain (1) can be transformed into frequency domain as

$$\mathfrak{L}(u(t)) = \mathfrak{L}\{K_P e(t) + K_I \int_0^t e(t)dt + K_D \frac{d}{dt} e(t)\} = \left(K_P + K_I \frac{1}{s} + K_D s\right) E(s). \tag{17}$$

where $E(s) = \frac{\theta^*}{s} - \theta(s)$. Compare to control system, control signal $u(t)$ is replaced to network parameter $\theta(t)$ in the deep neural network model. Then, we can rewrite the Equation (17) as

$$\theta(s) = \left(K_P + K_I \frac{1}{s} + K_D s\right) \left(\frac{\theta^*}{s} - \theta(s)\right). \tag{18}$$

Consequently, the standard closed-loop transfer function of LA-PID model is

$$E(s) = \frac{Y(s)}{X(s)} = \frac{1}{K_D} \cdot \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{19}$$

where

$$\begin{cases} (K_P + 1)/K_D = 2\xi\omega_n \\ K_I/K_D = \omega_n^2 \end{cases} \tag{20}$$

Then, we have

$$\xi = \frac{K_P + 1}{2\sqrt{K_I K_D}}, \quad \omega_n = \sqrt{\frac{K_I}{K_D}}. \tag{21}$$

As a result, the characteristic equation can be given as follows.

$$s^2 + \frac{K_P + 1}{K_D} s + \frac{K_I}{K_D} = 0 \tag{22}$$

In the principles of automatic control, the distribution of poles of the closed-loop characteristic equation in the complex plane (S-plane) plays a crucial role in determining the stability of second-order systems. The roots of the characteristic equation (22) can be solved, and these roots are related to the damping ratio ξ.

With the help of the dynamic characteristics of a second-order system, the convergence of network parameter $\theta(t)$ (4) can be systematically analyzed from **Case 1** to **Case 5**, where **Case 3**, **Case 4**, and **Case 5** are converged.

Case 1. Negative Damping: $\xi < 0$ (i.e., $\frac{K_P+1}{2\sqrt{K_I K_D}} \leq 0$)

In this case, the system has two real positive roots for the characteristic equation (22), and its unit step response can be written as

$$y(t) = 1 - \frac{\mathrm{e}^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} \sin(\omega_n \sqrt{1-\xi^2} t + \beta),\ t \geq 0.$$

where $\beta = \arctan\left(\sqrt{1-\xi^2}/\xi\right)$.

Since the damping ratio $\xi < 0$, the exponential factor has a positive power index $-\xi\omega_n t > 0$, so the dynamic process of the system is in the form of sinusoidal oscillation or monotonic divergence, indicating that the second-order system is *unstable* when $\xi < 0$.

Case 2. Undamped: $\xi = 0$ (i.e., $\frac{K_P+1}{2\sqrt{K_I K_D}} = 0$)

In this case, the characteristic equation (22) has a complex conjugate pair of imaginary roots, i.e.,

$$s_{1,2} = \pm j\omega_n.$$

This corresponds to a complex conjugate pair of poles along the imaginary axis in the S-plane.

And from its unit step response

$$y(t) = 1 - \cos(\omega_n t),\ t \geq 0.$$

it can be observed that the system's step response is characterized by continuous oscillation (*unstable*). In this scenario, the system is equivalent to an undamped condition.

Case 3. Underdamped: $0 < \xi < 1$ (i.e., $0 < \frac{K_P+1}{2\sqrt{K_I K_D}} < 1$)

At this time, the characteristic equation (22) has a pair of conjugate negative roots with negative real parts

$$s_{1,2} = -\xi\omega_n \pm j\omega_n\sqrt{1-\xi^2} = -\frac{K_P+1}{2K_D} \pm j\frac{\sqrt{4K_I K_D - (K_P+1)^2}}{2K_D}.$$

This result corresponds to complex conjugate poles situated in the left half-plane of the S-plane. And its unit step response is manifested as a damped oscillatory process, which can be written as

$$\begin{aligned} y(t) &= 1 - \frac{\mathrm{e}^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} \sin\left(\omega_n\sqrt{1-\xi^2}t + \arctan\frac{\sqrt{1-\xi^2}}{\xi}\right) \\ &= 1 - \frac{2\sqrt{K_I K_D}\mathrm{e}^{-\frac{K_P+1}{2K_D}t}}{\sqrt{4K_I K_D - (K_P+1)^2}} \sin\left(\frac{\sqrt{4K_I K_D - (K_P+1)^2}}{2K_D}t + \arctan\sqrt{\frac{4K_I K_D}{(K_P+1)^2} - 1}\right),\ t \geq 0. \end{aligned}$$

Under this result, we can derive the overshoot ($\sigma\%$), setting time (t_s) and oscillation period (T) to describe the dynamics of system (4).

$$\begin{cases} \sigma\% = \exp\left(-\frac{\pi}{\sqrt{\frac{4K_I K_D}{(K_P+1)^2}-1}}\right) \times 100\%, \\ t_s \approx \frac{6K_D}{K_P+1}, \\ T = \frac{2\pi}{\sqrt{\omega_n\sqrt{1-\xi^2}}}. \end{cases}$$

Case 4. Critically Damped: $\xi = 1$ (i.e., $\frac{K_P+1}{2\sqrt{K_I K_D}} = 1$)

In this case, the eigenvalues of the characteristic equation are

$$s_{1,2} = -\frac{K_P+1}{2K_D}.$$

From this result, the characteristic equation has two equal negative real roots, corresponding to two identical real poles located on the negative real axis of the S-plane.

And under the unit step response, the system output can be described as

$$y(t) = 1 - \mathrm{e}^{-\omega_n t}(1 + \omega_n t) = 1 - \mathrm{e}^{-\sqrt{\frac{K_I}{K_D}}t}\left(1 + \sqrt{\frac{K_I}{K_D}}t\right),\ t \geq 0.$$

From this mathematical terms, it can be observed that the step response asymptotically approaches a steady-state output without periodic oscillations.

Case 5. Overdamped: $\xi > 1$ (i.e., $\frac{K_P+1}{2\sqrt{K_I K_D}} > 1$)

The system characteristic equation (22) can be solved two distinct negative real roots in this situation.

$$s_{1,2} = -\xi\omega_n \pm \omega_n\sqrt{\xi^2 - 1} = -\frac{K_P+1}{2K_D} \pm \frac{\sqrt{(K_P+1)^2 - 4K_I K_D}}{2K_D}.$$

This pair of solutions corresponds to two distinct real poles located on the negative real axis of the S-plane. Furthermore, under the unit step response, the system output can be described as

$$y(t) = 1 + \frac{\mathrm{e}^{-t/T_1}}{T_2/T_1 - 1} + \frac{\mathrm{e}^{-t/T_2}}{T_1/T_2 - 1},\ t \geq 0.$$

where $T_1 = \frac{1}{\omega_n(\xi-\sqrt{\xi^2-1})}$, $T_2 = \frac{1}{\omega_n(\xi+\sqrt{\xi^2-1})}$, ω_n, ξ is defined in Equation(21).

It can be seen that the system corresponding unit step response also asymptotically approaches a steady-state output without periodic oscillations, but with a slower response rate compared to the critically damped case. □

C. Additional Experiments

Table C1. Test accuracy on 5-way classification for FC100 and CIFAR-FS.

	Backbone	**FC100**		**CIFAR-FS**	
		1-shot	5-shot	1-shot	5-shot
Random Init	4-CONV	27.50±0.45%	35.37±0.48%	29.74±0.46%	39.87±0.49%
LA-PID +Random Init		**32.42±0.46**%	**46.58±0.49**%	**38.28±0.45**%	**54.21±0.47**%
MAML	4-CONV	36.67±0.48%	49.38±0.49%	56.80±0.49%	74.97±0.43%
LA-PID +MAML		**46.23±0.47**%	**59.59±0.49**%	**64.23±0.47**%	**78.90±0.45**%
23L2TT+MAML	4-CONV	-	-	58.50±0.11%	76.16±0.09%
ALFA+MAML		37.99±0.48%	53.01±0.49%	59.96±0.49%	76.79±0.42%
Random Init	ResNet12	32.26±0.47%	42.00±0.49%	36.86±0.48%	49.46±0.50%
LA-PID +Random Init		**34.72±0.47**%	**45.52±0.49**%	**44.79±0.48**%	**57.48±0.48**%
MAML	ResNet12	37.92±0.48%	52.63±0.50%	64.33±0.48%	76.38±0.42%
LA-PID +MAML		**47.75±0.49**%	**60.84±0.48**%	**71.44±0.45**%	**85.15±0.35**%
23L2TT+MAML	ResNet12	-	-	73.63±0.11%	85.76±0.08%
ALFA+MAML		42.46±0.49%	55.82±0.50%	66.79±0.47%	83.62±0.37%