


# An Optimal Control View of LoRA and Binary Controller Design for Vision Transformers

Chi Zhang<sup>1</sup>, Jingpu Cheng<sup>1</sup>, and Qianxiao Li<sup>1</sup>

Department of Mathematics, National University of Singapore  
{czhang24, chengjingpu, qianxiao}@nus.edu.sg

**Abstract.** While recent advancements in model fine-tuning predominantly emphasize the utilization of low-rank adaptation (LoRA), we propose an alternative approach centered on reducing the precision of adaptation matrices. In particular, we depart from the common viewpoint that considers adaptation matrices solely as weight differences, and reinterpret them as “control variables” to perturb pre-trained ViT systems. This new perspective enables the establishment of a control-oriented framework, facilitating the exploration of optimal controls guided by the Pontryagin Maximum Principle. Furthermore, we demonstrate that for bounded control sets such as hypercubes, the optimal controls often take on boundary values, leading naturally to a binary controller design. Theoretical analysis reveals that employing a binary control strategy achieves the same reachable state as its full-precision counterpart in the continuous idealisation of deep residual structures, a finding corroborated by later empirical investigations. Our studies further indicate that the controller’s rank holds greater significance than its precision. As such, opting for low-precision yet high-rank controls is demonstrated to obtain better performance for practical vision tasks.

**Keywords:** Low-Rank Adaptation · Optimal Control

## 1 Introduction

Training a Vision Transformer (ViT) [17] from scratch often extends over a large corpus of data and consumes thousands of GPU hours. For practical applications, it is more favorable to refrain from such a resource-intensive training process. Instead, adapting a pre-trained model to downstream tasks proves to be more favorable. This adaptation process commonly employs the full-tuning mechanism, wherein all parameters of a pre-trained model undergo iterative adjustment to align with task-specific data.

One major challenge lies in the substantial number of parameters, a predicament that may be exacerbated by the increasing model capacity. To tackle this issue, recent research focuses on the study of Parameter-Efficient Fine-Tuning (PEFT), by either selectively updating a subset of parameters or injecting some new parameters [21, 39, 49]. In particular, the Low-Rank Adaptation (LoRA) algorithm [22], along with its vision-specific counterpart named AdaptFormer [11],

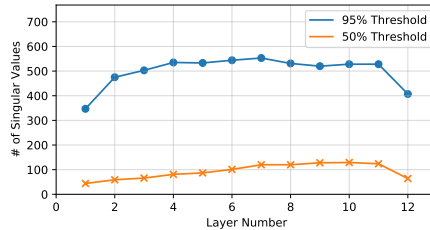
incorporates additional trainable dense layers featuring low-rank matrices while keeping the pre-trained ViT model fixed. Empirical investigations have demonstrated that, with very low-rank matrices (even as small as 1 rank), this approach can attain performance levels closely to its full-rank counterpart.

Despite its practical success, the fundamental mechanisms of LoRA remain to be fully understood. Existing explanations [1, 22] propose that weight changes  $\{\Delta W\}$  may exhibit “low intrinsic ranks”, hence advocating the use of low-rank matrices to approximate these changes. Yet, there is a lack of theoretical evidences that such ranks could be as minimal as one. In fact, subsequent studies indicate that achieving a comparable expressive power to full-tuning may necessitate a sufficient number of ranks in LoRA, for instance “embedding/2” in [56]. Empirical results also corroborate these findings, as depicted in Figure 1, indicating that a significant number of ranks may be necessary to accurately reconstruct  $\Delta W_{\text{MLP}}$  on MLP layers. Question remains why low-rank matrices could approximate such weight differences in practice. Moreover, considering the highly nonlinear nature of each ViT block, it is unclear why an (almost) linear low-rank updating scheme can replicate the nonlinear distinctions after model adaptation.

These gaps in understanding LoRA motivate our investigations into its underlying mechanisms. But instead of following the existing concepts, we introduce an alternative view: we no longer perceive  $\{\Delta W\}$  solely as weight disparities, but conceive them as *control variables* to perturb the pre-trained ViT systems. This interpretation draws analogies to the classical control theory [6, 18, 31], where controls are often introduced to guide a complex system (e.g., an airplane [44]) towards the desired state. In a parallel vein, we may leverage control variables to steer a pre-trained ViT system towards the minimal loss on downstream tasks.

This new perspective inspires us to search for the optimal controls. In particular, we formulate a necessary condition based on the Pontryagin’s Maximum Principle (PMP) [45], mandating the control variables to maximize the Hamiltonian function. This condition serves as a universal requirement applicable to controls on ViT of diverse natures, be they linear or nonlinear. The LoRA algorithm can be shown as a special case of linear control under the PMP condition, thereby enhancing its theoretical applicability.

Furthermore, the link to optimal control theory opens avenues for alternative forms of controller design. Specifically, the classical Bang-Bang theory enables the design of a binary control mechanism when maximizing the Hamiltonian



**Fig. 1:** Number of singular values to capture 50% and 95% of the total value. The findings suggest that, to recover the predominant  $\Delta W_{\text{MLP}}$  after adaptation on CIFAR-100, certain layers may require a rank exceeding 500.

function. Control matrices are then simplified to their binary essence, in order to minimize the computation/communication costs. Theoretical analysis suggests that such a binary control strategy has the same reachable states as its full-precision counterpart in continuous time. Consequently, we are able to design low-precision (and low-rank) controllers for practical model adaptation, which is later validated by a series of empirical experiments on vision tasks.

In summary, our contributions are three-fold. (1) Firstly, we bridge the study of modern PEFT algorithms with optimal control theory, by introducing a control-oriented framework for model adaptation (Section 3.1). (2) Within this framework, we present an optimal control interpretation for LoRA (Section 3.4), offering a new perspective that may circumvent several challenges inherent in existing understandings. (3) Lastly, we devise a binary control mechanism (Section 4) and prove that it has the same reachable state as the standard LoRA algorithm in continuous time. As such, the controller can be compressed to the binary forms, endowing it with computational and communicative efficiency in practical applications.

## 2 Related Works

**Parameter Efficient Fine-Tuning:** Tuning (vision) transformer models by adjusting only the head layers typically results in a notable decline in downstream performance. Consequently, recent studies of transfer learning have concentrated on the optimization of selecting a subset of parameters or introducing new lightweight parameters. In particular, the prompt-based algorithms [9,46,47] involve providing extra instructions or queries to the tokens, in order to guide the model’s behavior. Following works in [25,32,36,50] have demonstrated the effectiveness of such a prompt-based method for various NLP and CV downstream tasks. Alternatively, a cohort of studies [21,26,39] corroborates the viability of incorporating additional adapter layers. These auxiliary layers are selectively integrated into the existing architecture, allowing for task-specific adaptation without extensive retraining of the entire model. The concept of low-rank adaptation was initially introduced by [49], as a solution to reduce parameters for adapters. The following LoRA works [11,22,53,58] refined the tuning architecture, placing it in parallel with the original Vision Transformer (ViT) blocks rather than in a sequential order. Such a parallel low-rank scheme often yields lower inference latency and practical performance improvements [22]. Besides, when compared with prompt-based algorithms, the rank of adaptation matrices can be easily scaled up to obtain similar performance to full-tuning.

**Optimal Control in Machine Learning:** Optimal control [28,37], as a natural extension of the variational method [19], seeks to determine the trajectory of a dynamic system that minimizes or maximizes a certain performance criterion. A cornerstone to applications in deep learning may trace back to the seminar works [10,20,54], where authors treated machine learning as function approximation via a control system. Following this concept, neural networks have been interpreted as discretisations of an optimal control problem, subject to an ordi-

nary differential equation constraint in [5, 27, 61]. In particular, [34, 35] analyzed continuous-time analogues of neural networks in the optimal control framework and derived MSA-based algorithms. This paper draws analogies to these pioneering works and seeks the optimal controls, and focuses on the model tuning within the same theoretical framework.

**Quantized Optimization:** Our work develops a binary controller for model tuning in downstream tasks. A body of prior research [14, 23, 33, 38] has investigated the viability of binary or other quantized networks. The gradient optimization in this paper resembles the methodology in [63], where a trainable soft bounding  $c_t$  is introduced instead of using fixed  $\pm 1$  as binary values. This allows the control values within ViT to have more flexibility and controllability when applying to downstream tasks. Quantized parameters are also studied in model adaptation [16, 55], mainly for NLP tasks. But these works focus on quantizing the pre-trained model, while this paper aims to reduce the precision of controller.

### 3 An Optimal Control Framework for PEFT Algorithms

In this part, we develop a control-oriented view for perturbed ViT systems and derive the necessary condition for optimal controls.

#### 3.1 Dynamics of Controlled ViT Systems

To commence, we first introduce the dynamics of ViT systems with controls. Given the  $i$ -th image  $x_0^i \in \mathbb{R}^{C \times H \times W}$ , the ViT model first splits and embeds it into a sequence of tokens, denoted as  $\bar{x}_1^i \in \mathbb{R}^{d \times h}$ , where  $d$  represents the number of image tokens and  $h$  refers to the dimension of each token. Subsequently, an additional class-token  $x_{1,\text{cls}}^i$  is concatenated to these tokens, followed by adding a positional embedding to form the final  $x_1^i$ .

The LoRA algorithm [22] then focuses on processing these tokens with the following dynamics:

$$x_{n+1}^i = f_n(x_n^i, \theta_n, g_n(x_n^i, \mu_n)), \quad n \in [1, \dots, N-1]. \quad (1)$$

Here  $f_n$  represents the flow on the  $n$ -th pre-trained ViT block, typically consisting of a multi-head attention layer and a fully-connected MLP layer.  $g_n$  denotes a perturbation function that takes  $x_n^i$  as input and incorporates some control variables  $\mu_n$ . For instance, the controls take on a low-rank form in LoRA, expressed as  $\mu_n = A_{1,n}A_{2,n}$ , where  $A_{1,n} \in \mathbb{R}^{h \times r}$ ,  $A_{2,n} \in \mathbb{R}^{r \times h}$ , and  $r \ll d$ . For efficiency concerns, the pre-trained parameters  $\theta_n$  remain fixed throughout adaptation, while only the control parameters  $\mu_n$  are trainable.

The following AdaptFormer algorithm [11] relocates the low-rank matrices to be parallel to the ViT blocks. As such, the controlled dynamics become

$$x_{n+1}^i = f_n(x_n^i, \theta_n) + g_n(x_n^i, \mu_n), \quad n \in [1, \dots, N-1]. \quad (2)$$

One advantage of this formulation is that the control function  $g_n$  is detached from the original  $f_n$ , leading to simpler control analysis. Therefore we shall adhere to this formulation throughout the paper.

### 3.2 Challenges of Existing Explanations for LoRA

Such a controlled system has witnessed broad applications in transfer learning, yet its fundamental mechanism remains under-explored. Existing studies [1, 22] often posit that optimizing  $\mu_t$  aims to align the weight changes with those in full-tuning procedures. However, several critical gaps persist in this perspective. First of all, to faithfully replicate the weight changes in full-tuning, the approximation with low-rank matrices should be applied to *all parameters*, given that they are all trainable in model adaption. But empirical results often show that it is sufficient to apply LoRA to certain matrices, such as only the query  $W_Q$  and value  $W_K$  in self-attention. Moreover, as depicted in Figure 1, theoretical considerations suggest a relatively large rank requirement for adequately approximating weight changes, yet empirical evidences consistently show that low-rank matrices can achieve performance levels comparable to their full-rank counterparts. Finally, it remains intricate to understand why linear adaptation matrices can accurately replicate weight changes in non-linear ViT blocks, especially under the low rank setting.

### 3.3 Continuous-Time Formulation and Optimal Control with PMP

**TL;DR:** We consider  $g_n$  as a control function to perturb the original model  $f_n$ , and seek the optimal controls with the classical Pontryagin Maximum Principle from control theories.

To address the above challenges, we provide an optimal control formulation in this part. For the practical ViT models, the functions  $f_n$  in (2) actually takes a residual form, i.e.  $f_n(x_n, \theta_n) = x_n + r_n(x_n, \theta_n)$  for some function  $r_n$ . When  $N$  is large and the scale of  $r_n$  and  $g_n$  are small relative to  $x_n$ , (2) can be idealized as a continuous-time control system

$$\dot{x}_t = r_t(x_t, \theta_t) + g_t(x_t, \mu_t), \quad (3)$$

where  $t \in [0, T]$  (with  $T := N\delta$ ) represents the continuous counterpart of the indices for successive blocks. The functions  $x_t, r_t, g_t, \mu_t$  are the scaled continuations of  $x_n, r_n, g_n, \mu_n$  respectively, divided by a small factor  $\delta > 0$ .

The overall formulation in the continuous time could be expressed as:

$$\begin{aligned} \min_{\{\mu_t \in \mathbb{M}_t\}} \quad & \frac{1}{M} \sum_{i=1}^M L(x_T^i, y^i) \\ \text{s.t.} \quad & \dot{x}_t^i = r_t(x_t^i, \theta_t) + g_t(x_t^i, \mu_t), \quad t \in [0, T] \end{aligned} \quad (4)$$

where  $L(x_T^i, y^i) := \bar{L}(f_{\text{final}}(x_{T, \text{cls}}^i), y^i)$  for some loss function  $\bar{L}$  measures the difference between the model prediction and the true label, and  $\mathbb{M}_t$  denotes the admissible control set at time  $t$ .

One may recognize the above problem (4) as a classical fixed-time control problem. In particular, the objective is to find a sequence of optimal controls

$\{\mu_t^*\}$  to minimize the terminal loss  $L$ . A typical solution would be through the classical Pontryagin Maximum Principle (PMP) [45], which depicts the best possible control to transition a dynamical system from one state to another, especially in the presence of constraints.

To proceed, let us assign a series of co-state variables  $\{p_t^i \mid p_t^i \in \mathbb{R}^{d \times h}\}$ , which are of the same dimension as the states  $\{x_t^i\}$ . We can then define the Hamiltonian  $H_t : \mathbb{R}^{d \times h} \times \mathbb{R}^{d \times h} \times \mathbb{M}_t \rightarrow \mathbb{R}$  as

$$H_t(x, p, \mu) := \text{Tr}([r_t(x, \theta_t) + g_t(x, \mu)]p^T) \quad (5)$$

with  $\text{Tr}$  denoting the trace of a matrix. The following PMP theorem then defines a necessary condition for the optimal control.

**Theorem 1 (Pontryagin’s Maximum Principle [37]).** *Suppose that  $r_t$  and  $g_t$  are both continuous in  $t$  and continuously differentiable w.r.t.  $x$ , and  $L$  is differentiable in  $x$ . Denote  $\{\mu_t^*\}$  a bounded optimal control to (4), and  $\{x_t^{i,*}\}$  the corresponding optimally controlled state. Then, there exist optimal co-states  $\{p_t^{i,*}\}$  which are absolutely continuous in  $t$  such that the following Hamiltonian equations are satisfied:*

$$\dot{x}_t^{i,*} = \nabla_p H_t(x_t^{i,*}, p_t^{i,*}, \mu_t^*), \quad x_0^{i,*} = x_1^i, \quad (6)$$

$$\dot{p}_t^{i,*} = -\nabla_x H_t(x_t^{i,*}, p_t^{i,*}, \mu_t^*), \quad p_T^{i,*} = -\frac{1}{M} \nabla L(x_T^{i,*}). \quad (7)$$

Moreover, the optimal control variables  $\{\mu_t^*\}$  should satisfy the Hamiltonian maximization condition:

$$\sum_{i=1}^N H_t(x_t^{i,*}, p_t^{i,*}, \mu_t^*) \geq \sum_{i=1}^N H_t(x_t^{i,*}, p_t^{i,*}, \mu), \quad \forall \mu \in \mathbb{M}_t. \quad (8)$$

Solving the optimal controls in the above Hamiltonian maximization requires us to provide a concrete form of  $g_t$  in Eq (5). For parameter efficiency concerns, the AdaptFormer algorithm considers the following form  $g_n := \sigma(x_n^i A_{1,n}) A_{2,n} + B_n$ , where  $\sigma$  denotes an activation function like ReLU. The subsequent research [40] demonstrates that such an activation function has minimal effects in practice, making it safe to consider a pure linear form:  $g_n := x_n^i A_n + B_n$ .

In a similar vein, we may also consider the linear control formulation and require  $A_t \in \mathbb{A}_t \subseteq \mathbb{R}^{h \times h}$  and  $B_t \in \mathbb{B}_t \subseteq \mathbb{R}^{d \times h}$  for some control sets  $\mathbb{A}_t$  and  $\mathbb{B}_t$ . Note here we no longer consider  $A_t$  and  $B_t$  as the weight difference  $\Delta W_t$  of the frozen weights  $\theta_t$ , but view them as the control variables to minimize the terminal loss. By combining the linear control with Eq (5), we can obtain the Hamiltonian function  $H_t : \mathbb{R}^{d \times h} \times \mathbb{R}^{d \times h} \times \mathbb{A}_t \times \mathbb{B}_t \rightarrow \mathbb{R}$  for the linear case:

$$H_t(x, p, A, B) := \text{Tr}([r_t(x, \theta_t) + xA + B]p^T).$$

The PMP theorem mandates the optimal controls  $A_t^*$  and  $B_t^*$  to maximize such a Hamiltonian function. To obtain these optimal controls, one common

approach [34] involves a random choice of initial values  $A_t^0$  and  $B_t^0$ , and then iteratively updates these control variables in the following rule:

$$(A_t^l, B_t^l) = \arg \max_{A, B} \sum_i H_t(x_t^i, p_t^i, A, B) - \rho \|A - A_t^{l-1}\|^2 - \rho \|B - B_t^{l-1}\|^2, \quad (9)$$

where  $A_t^l, B_t^l$  represent the values of  $A_t, B_t$  in the  $l$ -th iteration. Note this argmax formulation can be solved in closed-form solutions, namely

$$A_t^l = A_t^{l-1} + \frac{1}{2\rho} \sum_i \sum_j [x_t^i]^T [p_t^i]_j, \quad (10)$$

$$B_t^l = B_t^{l-1} + \frac{1}{2\rho} \sum_i p_t^i, \quad (11)$$

where we denote the  $j$ -th row of instance  $x_t^i$  as  $[x_t^i]_j$ .

### 3.4 An Optimal Control Interpretation for LoRA

In the following, we show the above control-based updating rules (10) (11) can also be obtained by the gradient-descent step in LoRA in the discrete cases. By performing backward computation, we have  $p_N = -\frac{1}{M} \nabla_{x_N^i} L(x_N^i)$ . Then with the chain rule, we have:

$$\begin{aligned} \nabla_{A_n} \left( \frac{1}{M} \sum_i L(x_N^i) \right) &= \sum_i \frac{1}{M} \nabla_{x_{n+1}^i} L(x_N^i) \nabla_{A_n} x_{n+1}^i = - \sum_i p_{n+1}^i \nabla_{A_n} (x_n^i A_n) \\ &= - \sum_i \nabla_{A_n} \text{Tr}(x_n^i A_n (p_n^i)^T) = - \sum_i \sum_j [x_n^i]^T [p_{n+1}^i]_j. \end{aligned} \quad (12)$$

Similarly,

$$\nabla_{B_n} \left( \frac{1}{M} \sum_i L(x_N^i) \right) = - \sum_i p_{n+1}^i. \quad (13)$$

By examining the above equations, we show that the GD optimization (12)(13) for  $A_n$  and  $B_n$  are consistent with Eqs (10)(11), with a learning rate of  $\frac{1}{2\rho}$ .

Here we demonstrate that the GD optimization steps (12)(13) are aligned with optimal control rules in (10)(11). As such, the control view provides an alternative explanation for LoRA: the optimization of low-rank matrices may not be simulating the “weight differences”, but achieving optimal controls on downstream tasks. Such a new view avoids the above pitfalls in Section 3.2 and is more close to the real-world applications. For instance, it is generally unnecessary to apply controls to all parameters within a model. As a canonical example, for a robotic arm, it is often adequate to apply controls solely to specific joints to achieve desired positioning and orientation [15]. Similarly, controls may

be selectively applied to certain blocks of a ViT system, such as specific parts of an attention block or solely on the MLP layer. Furthermore, there is flexibility in choosing the forms of controls: it is often feasible to apply a linear control to a nonlinear block.

## 4 Binary Controller Design for ViT

Ever since June 1696, the study of optimal control has undergone a developmental journey over the past three centuries [52]. By bridging the study of modern efficient-tuning algorithms with classical optimal control theory, we are granted the capacities to utilize the classical control approaches to study transfer learning. A complete analysis is nevertheless beyond the scope of this paper, and we restrict our analysis to another category of linear controller rooted in the principle of Hamiltonian maximization in this section. Subsequently, we prove such a binary controller (BiC) yields an identical reachable set as the original LoRA method in the continuous time, even with only bit-wise precision.

### 4.1 Hamiltonian Maximization with Bang-Bang Control

In Eq (9), the Hamiltonian maximization is performed in a conservative way, with two supplementary regularization terms  $\|A - A_t^{l-1}\|^2$  and  $\|B - B_t^{l-1}\|^2$ . Alternatively, we may retain the original form of the Hamiltonian maximization:

$$(A_t^*, B_t^*) = \arg \max_{A, B} \sum_i H_t(x_t^i, p_t^i, A, B), \quad (14)$$

and endeavor to solve it in closed-form solutions.

**Lemma 1 (Bang-Bang Control).** *Let the control sets for  $A_t, B_t$  be  $\mathbb{A}_t = [-1, 1]^{h \times h}$  and  $\mathbb{B}_t = [-1, 1]^{d \times h}$ , respectively. Then, Eq (14) admits a closed-form solution as:*

$$[A_t^*]_{m,n} = \operatorname{sgn} \left( \sum_i \sum_j \left[ [x_t^{i,*}]_j^T [p_t^{i,*}]_j \right]_{m,n} \right), \quad (15)$$

$$[B_t^*]_{m,n} = \operatorname{sgn} \left( \sum_i \left[ p_t^{i,*} \right]_{m,n} \right), \quad (16)$$

where  $\operatorname{sgn}$  refers to the sign operation and  $[*]_{m,n}$  denotes the  $(m, n)$ -th element. If the expression within the brackets of the sign functions in Eqs. (15) or (16) equals zero, then the corresponding argmax is arbitrary.

Despite the potential to take any values from the interval  $[-1, 1]$ , the optimal control strategy in the above lemma involves taking the extreme values in the feasible set. Such a phenomenon, generally known as the ‘‘Bang-Bang Control’’,

frequently arises in classical optimal control problems [2, 4, 51]. For example, if a driver wants to reach another city in the shortest time, the optimal solution entails applying the maximum acceleration (maximum control) during the trip and subsequently employing maximum braking (minimum control) before arrival.

Optimal controls for the ViT systems adhere to the same principle, wherein the optimal control matrices  $A_t^*$  and  $B_t^*$  are anticipated to consist of only binary values, namely  $\pm 1$ . In more general scenarios where the control is constrained within the range  $[-c, c]$ , the optimal control is expected to manifest as  $\pm c$ .

## 4.2 Performance Analysis

A fundamental question arises as to whether the utilization of binary controls would compromise the system’s expressive ability. To address this question, we conduct the following theoretical analysis.

For given control trajectory  $\mathbf{A} := \{A_t\}$ ,  $\mathbf{B} := \{B_t\}$ , the flow map

$$\varphi_{\mathbf{A}, \mathbf{B}} := x(0) \rightarrow x(T), \quad \text{where } \dot{x}(t) = r_t(x, \theta_t) + A_t x_t + B_t \quad (17)$$

characterizes the input-output relationship in the encoder segments of the idealized network. Our binary approach to the control design requires that, at each moment  $t \in [0, T]$ , the entries of  $A_t$  and  $B_t$  take binary values. The following theorem indicates that in this continuous-limit, these binary controls have essentially the same reachable set as its floating-point counterpart.

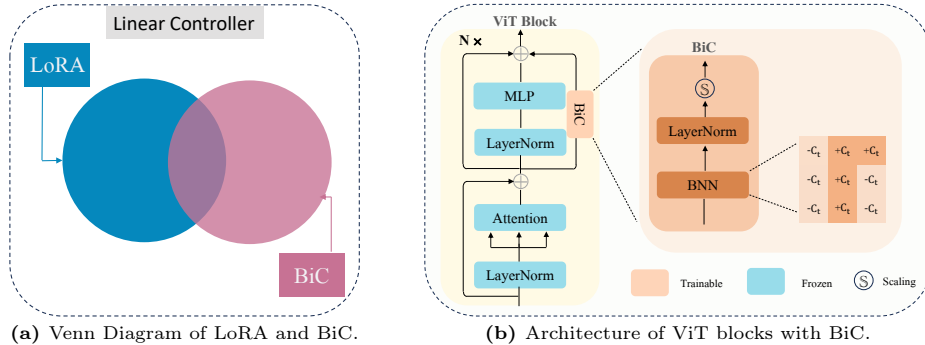
**Theorem 2 (Informal).** *Assume  $r_t(x, \theta_t)$  is Lipschitz continuous in  $x$ . Then, for any given full-precision controls  $\mathbf{A}, \mathbf{B}$  of (3), compact set  $K$  and  $\varepsilon > 0$ , there exists control  $\mathbf{A}_{\text{binary}}$ , and  $\mathbf{B}_{\text{binary}}$  that takes only binary values for each  $t \in [0, T]$ , such that*

$$\|\varphi_{\mathbf{A}, \mathbf{B}} - \varphi_{\mathbf{A}_{\text{binary}}, \mathbf{B}_{\text{binary}}}\|_{C(K)} < \varepsilon, \quad (18)$$

where  $\|\cdot\|_{C(K)}$  denotes the uniform norm on  $K$ .

Rigorous statements and the proof of Theorem 2 are detailed in Appendix A. The key intuition is that every matrix is a convex combination of binary matrices, i.e. for any full-precision matrix  $A \in \mathbb{R}^{d \times h}$ , there exists binary matrices  $A_1, \dots, A_N$  such that  $A = \sum_{i=1}^N \lambda_i A_i$  for some  $\lambda_i \geq 0$  and  $\sum_i \lambda_i = 1$ . Consequently, within a small interval  $[t, t + \Delta t]$ , we can unfold the convex combination of  $\sum_{i=1}^N \lambda_i A_i$  over time to approximate full-precision LoRA blocks with their binary version.

Theorem 2 indicates that in the case when the inputs change continuously through the layers of a ViT model, the utilization of only binary control will not lead to a drop in performance. Although extending this theoretical analysis to general discrete networks is more complex, insights from the idealized case indicate that binary control could potentially approximate the effects of full-precision controls by averaging their impact across layers.



**Fig. 2:** Both LoRA and BiC can be viewed as special cases of linear controllers. Details of how to utilize BiC to control ViT blocks is plotted on the right.

### 4.3 Binary Controller Design and Optimization

As the optimal strategy involves only extreme values, we may move towards designing a pure binary control (BiC) on pre-trained ViT systems. By doing so, we can replace floating-point controls with integer variables (or integers with an amplifier  $c$ ), thereby enhancing computational efficiency.

**Design:** We consider the soft bounding cases of confining our controls  $A_t$  within  $\{-c_t, c_t\}$ , where  $c_t$  is a layer-wise trainable scalar. Each element within the control matrices  $A_t$  may only choose  $c_t$  or  $-c_t$ . Similar to LoRA, the binary control matrices  $\{A_t\}$  may also possess the low-rank property, leading to low-rank and low-precision controls. In Figure 2a, such controls lie in the intersection of LoRA and BiC. Similarly, binary quantization can be applied to the bias matrix  $B_t$ . This enables the creation of trainable control matrices that exclusively consist of binary values. In particular, the control parameters  $\{A_t\}$   $\{B_t\}$  have the form  $A_t = c_{1,t}\bar{A}_t$ ,  $B_t = c_{2,t}\bar{B}_t$ , where  $c_{1,t}$  and  $c_{2,t}$  are full-precision scalars, and the entries of  $\{\bar{A}_t\}$  and  $\{\bar{B}_t\}$  all belong to  $\{+1, -1\}$ . The Hamiltonian function  $H_t$  is:

$$H_t(x, p, c_1, c_2, \bar{A}, \bar{B}) = \text{Tr} \left( [f_t(x, \theta_t) + c_1 x \bar{A} + c_2 \bar{B}] p^T \right). \quad (19)$$

Following the common practice in binary neural networks [35, 48, 62]<sup>1</sup>, we incorporate a normalization layer after binary control matrices. Existing norm methods like BatchNorm [24] or LayerNorm [3] can only normalize values to the range  $[0, 1]$ , and we anticipate perturbations in the system to be relatively small. To accommodate this, we introduce a trainable scaling value  $s_t \in \mathbb{R}$  to modulate the overall scale. The complete controlled ViT block is depicted in Figure 2b.

**Optimization**<sup>2</sup>: In general, the optimal control may be sought through the method of successive approximation (MSA) [13, 41]: we randomly guess an initial controls  $\{c_{1,t}^0\}$ ,  $\{c_{2,t}^0\}$ ,  $\{A_t^0\}$ ,  $\{B_t^0\}$  for  $\{c_{1,t}\}$ ,  $\{c_{2,t}\}$ ,  $\{A_t\}$ ,  $\{B_t\}$ , respectively.

<sup>1</sup> It is possible to remove the normalization layer as [8, 12]. We retain such a block in this paper for simplicity of implementations.

<sup>2</sup> We also provide a pure gradient method in Appendix C.

Subsequently, we compute the associated  $\{x_t^{i,0}\}$  and  $\{p_t^{i,0}\}$  corresponding to the prescribed dynamics (6)(7). Throughout each training epoch  $l$ , we update the parameters  $c_{1,t}, c_{2,t}, \bar{A}_t, \bar{B}_t$  by maximizing the Hamiltonian with penalty terms in  $\|c_{1,t}^{l+1} - c_{1,t}^l\|^2$  and  $\|c_{2,t}^{l+1} - c_{2,t}^l\|^2$ , i.e. for each  $t$  and epoch  $l$ , we set

$$(c_{1,t}^{l+1}, c_{2,t}^{l+1}, \bar{A}_t^{l+1}, \bar{B}_t^{l+1}) = \arg \max_{c_1, c_2, \bar{A}, \bar{B}} \left[ \sum_i H_t(x_t^{i,l}, p_t^{i,l}, c_1, c_2, \bar{A}, \bar{B}) - \rho_{1,t}^l \|c_1 - c_{1,t}^l\|^2 - \rho_{2,t}^l \|c_2 - c_{2,t}^l\|^2 \right], \quad (20)$$

where  $\rho_{1,t}^l, \rho_{2,t}^l > 0$  are penalty parameters. In practice, we may solve this maximization problem by seeking the closed-form solutions for  $c_{1,t}$  and  $c_{2,t}$  first,

$$c_{1,t}^{l+1} = c_{1,t}^l + \frac{1}{2\rho_{1,t}^l} \sum_i \text{Tr} \left( [x_t^{i,l} \bar{A}_t^l] [p_t^{i,l}]^T \right),$$

$$c_{2,t}^{l+1} = c_{2,t}^l + \frac{1}{2\rho_{2,t}^l} \sum_i \text{Tr} \left( [\bar{B}_t^l] [p_t^{i,l}]^T \right).$$

And subsequently update the binary control variables as

$$[\bar{A}_t^{l+1}]_{m,n} = \text{sgn} \left( c_{1,t}^{l+1} \sum_i \sum_j \left[ [x_t^{i,l}]_j^T [p_t^{i,l}]_j \right]_{m,n} \right), \quad (21)$$

$$[\bar{B}_t^{l+1}]_{m,n} = \text{sgn} \left( c_{2,t}^{l+1} \sum_i \left[ p_t^{i,l} \right]_{m,n} \right). \quad (22)$$

A potential issue of the above updating scheme (21) (22) is the unstable switching of the parameters between  $\pm 1$ . To mitigate this issue, we may either penalize the flipping with a regularization term [35] or flip the signs based on the momentum [63]. We opt for the latter approach, wherein  $c_{1,t}, c_{2,t}$  are clipped to be positive, and also introduce two auxiliary matrices  $\tilde{A}_t, \tilde{B}_t$ , initialized as  $\tilde{A}_t^0 = \bar{A}_t^0, \tilde{B}_t^0 = \bar{B}_t^0$ . In each training epoch, we update  $\tilde{A}_t$  and  $\tilde{B}_t$  by

$$[\tilde{A}_t^{l+1}]_{m,n} = [\tilde{A}_t^l]_{m,n} + \lambda_1 \sum_i \sum_j \left[ [x_t^{i,l}]_j^T [p_t^{i,l}]_j \right]_{m,n}, \quad (23)$$

$$[\tilde{B}_t^{l+1}]_{m,n} = [\tilde{B}_t^l]_{m,n} + \lambda_2 \sum_i \left[ p_t^{i,l} \right]_{m,n}. \quad (24)$$

where  $\lambda_1, \lambda_2 > 0$  are two small constants. Subsequently, the binary matrices are updated by taking the sign of the accumulated gradients in  $\tilde{A}_t$  and  $\tilde{B}_t$ :

$$[\bar{A}_t^{l+1}]_{m,n} = \text{sgn}([\tilde{A}_t^{l+1}]_{m,n}), \quad [\bar{B}_t^{l+1}]_{m,n} = \text{sgn}([\tilde{B}_t^{l+1}]_{m,n}). \quad (25)$$

In comparison to the previous rule only on the current gradients, such a scheme enhances stability by integrating the gradient over successive epochs.

**Table 1:** Comparison of algorithm performance, with AdaptFormer [11] acting as a vision-specific LoRA algorithm. We follow [11] to only report one-time accuracy, while repeated experiments on BiC are available in Appendix D.2. [†] lr is decreased by 0.1 on CIFAR-100. [‡] Bytes are computed on CIFAR-100.

Algorithm	Total Bytes <sup>‡</sup>	CIFAR-100	SVHN	Food-101
Full-Tuning	100%	87.90 <sup>†</sup>	97.67	90.09
Linear-Probing	0.08%	69.83	66.91	69.74
VPT	0.09%	82.44	94.02	82.98
AdaptFormer-64	1.46%	85.90	96.89	87.61
AdaptFormer-768	8.40%	86.73	97.04	88.39
BiC-64	0.13%	85.74	97.01	87.46
BiC-768	0.35%	86.61	97.08	88.31

## 5 Experiments

In this section, we focus on the validation of the binary control (BiC), by comparing it with a series of efficient-tuning algorithms.

### 5.1 Preliminary

*Experiment Settings and Baseline Methods.* For fair comparison, we mirror the experimental setting in [11]. For layer initialization, we set the scaling value  $s_t = 0$ , while retaining the rest as default in PyTorch. This resembles the general setting of LoRA, where the uppermost layer within each block is initialized with zeros. We compare our low-precision adaptation algorithm, denoted as BiC, with a few commonly used fine-tuning algorithms: 1) Full-tuning: all parameters are trainable; 2) Linear probing: appending an additional trainable linear layer on top of the pre-trained model while keeping the rest parameters fixed.; 3) Visual Prompt Tuning (VPT) [25]: concatenating a set of trainable tokens with existing image tokens; 4) AdaptFormer [11]: a vision specific LoRA algorithm by injecting full-precision trainable low-rank matrices to ViT systems.

### 5.2 Full-Precision and Binary Control Comparison

We commence our investigations with series of experiments on the CIFAR-100 [30], SVHN [43] and Food-101 [7] datasets, and our primary goal is to compare the performance of various fine-tuning algorithms.

Our previous analysis in Sec 4.2 substantiates the possibilities of employing binary controls in ViT systems, and experimental observations corroborate such a theoretical analysis. In particular, the binary control strategy on ViT systems yields comparable performance levels to its full-precision counterpart across all datasets. Moreover, the BiC-64 algorithm, serving as a *low-rank and low-precision* algorithm, exhibits slightly inferior performance when compared to the full-precision AdaptFormer-64 on the CIFAR-100 and Food-101 datasets.

**Table 2:** Comparison of algorithm performance under the same computation cost.

Algorithm	CIFAR-100	SVHN	Food-101
AdaptFormer-1	83.52	93.04	83.64
BiC-32	85.37 (+1.85)	97.03 (+3.99)	86.20 (+2.56)
AdaptFormer-4	84.83	96.19	85.42
BiC-128	86.29 (+1.46)	97.20 (+1.01)	87.87 (+2.45)

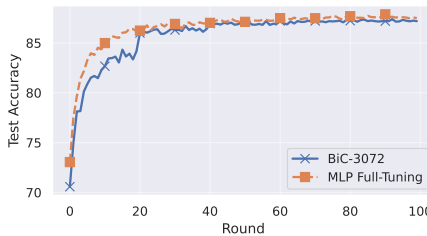
Interestingly, it demonstrates a tendency to outperform the full-precision algorithm on the SVHN dataset. Such a phenomenon has also been observed in the previous quantization study [63], where authors attributed to the capabilities of preventing overfitting.

### 5.3 Low Rank or Low Precision?

In practical applications, we often opt for light-weight controls. In pursuit of this objective, we have two orthogonal directions: reduce either rank or precision, as presented in Table 2. Unlike the seminal LoRA research [22] in NLP tasks, where authors have demonstrated that even a rank-1 matrix can serve as an effective adapter, vision downstream tasks are more sensitive to the reduction of rank. Specifically, the rank-1 AdaptFormer experiences significant accuracy degradation (e.g., a drop of 6.45% on the Food-101 dataset) when compared to the results of full-tuning in Table 1.

Alternatively, a strategy may opt to maintain the high ranks but sacrifice its precision, as exemplified by the BiC algorithm. Empirical results consistently show that this approach yields superior performance when compared to the low-rank counterpart across all datasets. Notably, in the case of BiC-32, we observe a substantial margin of improvement (e.g., +3.99%) when compared to the rank-1 LoRA algorithm on the SVHN dataset. Moreover, we explore the possibility of designing a high-rank binary control strategy in Figure 3. With a bit abuse of the concept

“rank”, we introduce a binary high-rank control (BiC-3072) for ViT systems and inserting additional activation function between the down and up projections. Results indicate that the bit-wise controls reach a performance level (87.39%) comparable to full-tune the MLP layers, and even surpass AdaptFormer-768 in



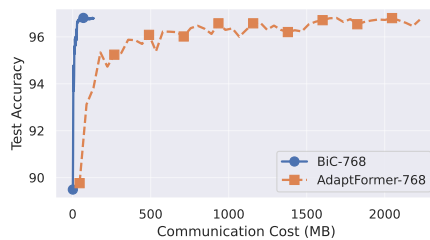
**Fig. 3:** Compare full-tune MLP layers with the utilization of BiC on the CIFAR-100 dataset. Result indicates that BiC exhibits a performance that is marginally lower, with a difference of merely 0.11%.

Table 1. Note the conventional LoRA algorithm is not suitable for this scenario, as it incurs the same computational cost as full-tune the entire MLP layers.

#### 5.4 Communication-Efficient ViT Adaptation

In addition to the low computational overhead, binary controls also find practical applications in the distributed optimization, such as Federated Learning [29, 42, 59]. In these cases, the server and client can download the pre-trained ViT models independently and only communicate the controllers. This also avoids the potential data leakage issues [57, 60] when transmitting the whole model. Owing to their inherently bit-wise nature, binary controllers typically result in a substantial reduction in communication costs.

To validate this property, we distribute the SVHN dataset to 5 nodes and measure the algorithm performance in Figure 4. Note LoRA can already effectively reduce the communication overhead by communicating only the adaptation matrices, and such a reduction can be further enhanced with binary controllers. Experiments demonstrate that, to achieve the same terminal test accuracy, binary controls incur only 3.5% communication cost of its full-precision counterpart on SVHN, rendering BiC as a highly communication-efficient algorithm in distributed optimization.



**Fig. 4:** Distributed algorithm performance on the SVHN dataset. BiC requires 30x less communication costs to reach the terminal accuracy on the SVHN dataset.

## 6 Conclusion

In this paper, we establish a connection between recent studies on PEFT algorithms like LoRA and classical optimal control theory. A control-oriented framework is introduced for efficient tuning, wherein adaptation matrices are reconceptualized as controls to perturb pre-trained ViT systems. Such a new perspective motivates us to define a necessary condition based on the Pontryagin Maximum Principle, where the LoRA algorithm can be recast as a special case of linear control. Furthermore, by maximizing the Hamiltonian function in a non-conservative way, we introduce a control strategy involving bit-wise adaptation matrices. Theoretical analysis affirms that this binary control strategy yields similar reachable states as its continuous counterpart, which is later validated in the empirical studies. Moreover, in scenarios where simplicity in controls is essential, our investigations suggest that the rank of the adaptation matrices are more important than their precision. Consequently, opting for low-precision yet high-rank controls proves to be more favorable in practical applications like distributed optimization.

## Acknowledgements

This research is supported by the National Research Foundation, Singapore, under the NRF fellowship (Project No. NRF-NRFF13-2021-0005).

## References

1. Aghajanyan, A., Zettlemoyer, L., Gupta, S.: Intrinsic dimensionality explains the effectiveness of language model fine-tuning. arXiv preprint arXiv:2012.13255 (2020)
2. Artstein, Z.: Discrete and continuous bang-bang and facial spaces or: look for the extreme points. *Siam Review* **22**(2), 172–185 (1980)
3. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
4. Bellman, R., Glicksberg, I., Gross, O.: On the “bang-bang” control problem. *Quarterly of Applied Mathematics* **14**(1), 11–18 (1956)
5. Benning, M., Celledoni, E., Ehrhardt, M.J., Owren, B., Schönlieb, C.B.: Deep learning as optimal control problems: Models and numerical methods. arXiv preprint arXiv:1904.05657 (2019)
6. Bishop, R.C.D.R.H.: *Modern control systems* (2011)
7. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101—mining discriminative components with random forests. In: *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part VI* 13. pp. 446–461. Springer (2014)
8. Brock, A., De, S., Smith, S.L., Simonyan, K.: High-performance large-scale image recognition without normalization. In: *International Conference on Machine Learning*. pp. 1059–1071. PMLR (2021)
9. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. *Advances in neural information processing systems* **33**, 1877–1901 (2020)
10. Chang, B., Meng, L., Haber, E., Ruthotto, L., Begert, D., Holtham, E.: Reversible architectures for arbitrarily deep residual neural networks. In: *Proceedings of the AAAI conference on artificial intelligence*. vol. 32 (2018)
11. Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., Luo, P.: Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems* **35**, 16664–16678 (2022)
12. Chen, T., Zhang, Z., Ouyang, X., Liu, Z., Shen, Z., Wang, Z.: “bn-bn=?”: Training binary neural networks without batch normalization. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 4619–4629 (2021)
13. Chernousko, F.L., Lyubushin, A.: Method of successive approximations for solution of optimal control problems. *Optimal Control Applications and Methods* **3**(2), 101–114 (1982)
14. Courbariaux, M., Bengio, Y., David, J.P.: Binaryconnect: Training deep neural networks with binary weights during propagations. *Advances in neural information processing systems* **28** (2015)
15. Craig, J.J.: *Introduction to robotics*. Pearson Educacion (2006)
16. Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L.: Qlora: Efficient finetuning of quantized llms. arXiv preprint arXiv:2305.14314 (2023)

17. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
18. Franklin, G.F., Powell, J.D., Emami-Naeini, A., Powell, J.D.: Feedback control of dynamic systems, vol. 4. Prentice hall Upper Saddle River (2002)
19. Gelfand, I.M., Silverman, R.A., et al.: Calculus of variations. Courier (2000)
20. Haber, E., Ruthotto, L.: Stable architectures for deep neural networks. *Inverse problems* **34**(1), 014004 (2017)
21. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: International Conference on Machine Learning. pp. 2790–2799. PMLR (2019)
22. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
23. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Binarized neural networks. *Advances in neural information processing systems* **29** (2016)
24. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. pmlr (2015)
25. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: European Conference on Computer Vision. pp. 709–727. Springer (2022)
26. Karimi Mahabadi, R., Henderson, J., Ruder, S.: Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems* **34**, 1022–1035 (2021)
27. Kerimkulov, B., Šiška, D., Szpruch, L.: A modified msa for stochastic control problems. *Applied Mathematics & Optimization* pp. 1–20 (2021)
28. Kirk, D.E.: Optimal control theory: an introduction. Courier Corporation (2004)
29. Konečný, J., McMahan, B., Ramage, D.: Federated optimization: Distributed optimization beyond the datacenter. arXiv preprint arXiv:1511.03575 (2015)
30. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
31. Kwakernaak, H., Sivan, R.: Linear optimal control systems, vol. 1. Wiley-interscience New York (1972)
32. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691 (2021)
33. Li, F., Liu, B., Wang, X., Zhang, B., Yan, J.: Ternary weight networks. arXiv preprint arXiv:1605.04711 (2016)
34. Li, Q., Chen, L., Tai, C., Weinan, E.: Maximum principle based algorithms for deep learning. *Journal of Machine Learning Research* **18**(165), 1–29 (2018)
35. Li, Q., Hao, S.: An optimal control approach to deep learning and applications to discrete-weight neural networks. In: International Conference on Machine Learning. pp. 2985–2994. PMLR (2018)
36. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190 (2021)
37. Liberzon, D.: Calculus of variations and optimal control theory: a concise introduction. Princeton university press (2011)
38. Lin, M., Ji, R., Xu, Z., Zhang, B., Wang, Y., Wu, Y., Huang, F., Lin, C.W.: Rotated binary neural network. *Advances in neural information processing systems* **33**, 7474–7485 (2020)

39. Lin, Z., Madotto, A., Fung, P.: Exploring versatile generative language model via parameter-efficient transfer learning. arXiv preprint arXiv:2004.03829 (2020)
40. Luo, G., Huang, M., Zhou, Y., Sun, X., Jiang, G., Wang, Z., Ji, R.: Towards efficient visual adaption via structural re-parameterization. arXiv preprint arXiv:2302.08106 (2023)
41. Lyubushin, A.: Modifications of the method of successive approximations for solving optimal control problems. *USSR Computational Mathematics and Mathematical Physics* **22**(1), 29–34 (1982)
42. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: *Artificial intelligence and statistics*. pp. 1273–1282. PMLR (2017)
43. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)
44. Nguyen, N.T., Nguyen, N.T.: *Model-reference adaptive control*. Springer (2018)
45. Pontryagin, L.S.: *Mathematical theory of optimal processes*. Routledge (2018)
46. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)
47. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. *OpenAI blog* **1**(8), 9 (2019)
48. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. In: *European conference on computer vision*. pp. 525–542. Springer (2016)
49. Rebuffi, S.A., Bilen, H., Vedaldi, A.: Learning multiple visual domains with residual adapters. *Advances in neural information processing systems* **30** (2017)
50. Sohn, K., Chang, H., Lezama, J., Polania, L., Zhang, H., Hao, Y., Essa, I., Jiang, L.: Visual prompt tuning for generative transfer learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 19840–19851 (2023)
51. Sonneborn, L., Van Vleck, F.: The bang-bang principle for linear control systems. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control* **2**(2), 151–159 (1964)
52. Sussmann, H.J., Willems, J.C.: 300 years of optimal control: from the brachystochrone to the maximum principle. *IEEE Control Systems Magazine* **17**(3), 32–44 (1997)
53. Touvron, H., Vedaldi, A., Douze, M., Jégou, H.: Fixing the train-test resolution discrepancy. *Advances in neural information processing systems* **32** (2019)
54. Weinan, E.: A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics* **1**(5), 1–11 (2017)
55. Xu, Y., Xie, L., Gu, X., Chen, X., Chang, H., Zhang, H., Chen, Z., Zhang, X., Tian, Q.: Qa-lora: Quantization-aware low-rank adaptation of large language models. arXiv preprint arXiv:2309.14717 (2023)
56. Zeng, Y., Lee, K.: The expressive power of low-rank adaptation. arXiv preprint arXiv:2310.17513 (2023)
57. Zhang, C., Ekanut, S., Zhen, L., Li, Z.: Augmented multi-party computation against gradient leakage in federated learning. *IEEE Transactions on Big Data* (2022)
58. Zhang, C., Jingpu, C., Xu, Y., Li, Q.: Parameter-efficient fine-tuning with controls. In: *Forty-first International Conference on Machine Learning* (2024)
59. Zhang, C., Li, Q.: Distributed optimization for degenerate loss functions arising from over-parameterization. *Artificial Intelligence* **301**, 103575 (2021)

60. Zhang, C., Xiaoman, Z., Sotthiwat, E., Xu, Y., Liu, P., Zhen, L., Liu, Y.: Generative gradient inversion via over-parameterized networks in federated learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5126–5135 (2023)
61. Zhang, D., Zhang, T., Lu, Y., Zhu, Z., Dong, B.: You only propagate once: Accelerating adversarial training via maximal principle. *Advances in Neural Information Processing Systems* **32** (2019)
62. Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., Zou, Y.: Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv preprint arXiv:1606.06160 (2016)
63. Zhu, C., Han, S., Mao, H., Dally, W.J.: Trained ternary quantization. arXiv preprint arXiv:1612.01064 (2016)