Parameter-Efficient Fine-Tuning with Controls

Chi Zhang [*1] **Jingpu Cheng** [*1] **Yanyu Xu** [2] **Qianxiao Li** [1]

Abstract

In contrast to the prevailing interpretation of Low-Rank Adaptation (LoRA) as a means of simulating weight changes in model adaptation, this paper introduces an alternative perspective by framing it as a control process. Specifically, we conceptualize lightweight matrices in LoRA as control modules tasked with perturbing the original, complex, yet frozen blocks on downstream tasks. Building upon this new understanding, we conduct a thorough analysis on the controllability of these modules, where we identify and establish sufficient conditions that facilitate their effective integration into downstream controls. Moreover, the control modules are redesigned by incorporating nonlinearities through a parameter-free attention mechanism. This modification allows for the intermingling of tokens within the controllers, enhancing the adaptability and performance of the system. Empirical findings substantiate that, without introducing any additional parameters, this approach surpasses the LoRA algorithms across all assessed datasets and rank configurations.

1. Introduction

Large-scale deep neural networks, in particular Transformers (Vaswani et al., 2017), have demonstrated unprecedented performance in areas such as computer vision (Dosovitskiy et al., 2020), natural language processing (Vaswani et al., 2017) and speech recognition (Radford et al., 2023). The prevailing methodology for attaining optimal model performance typically entails pre-training with an extensive text or image corpus, followed by subsequent fine-tuning on a compact task-specific dataset. For example, the Vision Transformer (ViT) (Dosovitskiy et al., 2020) is first pre-trained on the Imagenet-21K (Deng et al., 2009) or JFT (Sun et al., 2017) dataset, and then undergoes fine-tuning to facilitate adaptation on various downstream tasks.

The downside of such an approach is that the large models often consist of millions or even billions of parameters, leading to the exhaustive consumption of GPU memories during the full-tuning process. To alleviate this issue, a series of parameter-efficient fine tuning (PEFT) algorithms (Bapna et al., 2019; Houlsby et al., 2019; Jia et al., 2022; Lian et al., 2022) have been proposed in recent studies. One such example is the Low-Rank Adaptation (LoRA) (Hu et al., 2021), which integrates pairs of additional trainable low-rank matrices whilst keeping the original model fixed during the training process.

Despite its broad success in practical scenarios, the underlying mechanism of these PEFT algorithms remains underexplored. For instance, the general rationale behind LoRA algorithms is often framed as "the change in weights during model adaptation has a low intrinsic rank" (Aghajanyan et al., 2020; Hu et al., 2021). Consequently, to accurate replicate weight changes, low-rank matrices should be applied to *all parameters* within the model, given that they are all trainable in model adaptation. Yet, empirical results frequently suggest that it is sufficient to apply LoRA solely to specific matrices within each block, such as only the Query matrices Q_K.

In particular, we notice the recent AdaptFormer algorithm (Chen et al., 2022) opens the gate for a new direction. Unlike traditional approaches where LoRA matrices are nested within the attention blocks, AdaptFormer positions these low-rank matrices in parallel to the feed-forward layer. This arrangement allows for an alternative interpretation of PEFT algorithms in this paper: the LoRA matrices should not be merely regarded as the weight differences after model adaptation, but rather as control modules designed to perturb the original model.

Such an interpretation draws parallels to classical control theory (Bishop, 2011; Kwakernaak & Sivan, 1972), wherein light-module controls are frequently positioned to steer a complex system towards desired states. In the field of robotics (Slotine & Li, 1987; Lewis et al., 2003), for example, a controller may be employed to govern the movements of a robotic arm, orchestrating its precise positioning and motion to accomplish specific tasks. Analogously, transfer

[*]Equal contribution [1]Department of Maths, National University of Singapore, Singapore [2]The Joint SDU-NTU Research Center of Artificial Intelligence, Shandong University, China. Correspondence to: Qianxiao Li <Qianxiao@nus.edu.sg>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

learning for large models could be performed in a similar way: a series of lightweight modules (e.g., low-rank matrices) can be designed to control the ViT model, aiming to achieve minimal adaptation loss.

Following this new interpretation, we study the controllability of these low-rank modules in a general setting of the continuous-time analogue of multi-layer models (E, 2017; Haber & Ruthotto, 2017). Specifically, we establish sufficient conditions by necessitating the controller to span the space with full rank at any given time. Adhering to these conditions, we illustrate that the existing (almost) linear controls employed by AdaptFormer may face challenges in achieving full controllability, particularly when dealing with large datasets. Moreover, when considering transformers that incorporate attention mechanisms, our analysis indicates that there exist cases where the absence of cross-patch dynamics may lead to the failures of control.

These theoretical analyses further motivate us to devise new nonlinear controllers tailored for transformers on downstream tasks. To achieve this goal, we delve into the underlying mechanism of the original ViT model and propose a new nonlinear control featuring a parameter-free attention mechanism. Such a design allows the patches to be intermingled within controllers, thereby contributing to an enhancement in the overall controllability of the low-rank modules. Empirical findings consistently support that, without introducing any additional parameters, this approach outperforms the existing LoRA-like algorithms by a large margin on all assessed datasets and rank configurations.

In summary, our contributions encompass two key aspects. (1) Following the control-oriented perspective, we establish sufficient conditions for the controllability of perturbation functions. We scrutinize existing algorithms from this control viewpoint and demonstrate that they may fail to meet these conditions in certain cases. (2) In response to this, we redesign the controller module to align with the transformer architecture by incorporating a cross-patch attention mechanism. Numerical verification confirms that the proposed algorithm satisfies the controllability condition, and its effectiveness is further demonstrated across multiple datasets.

2. Related Works

Parameter-Efficient Fine-Tuning With the emergence of large-scale deep neural networks, transfer learning (Pan & Yang, 2009) with pre-trained models has become the de facto approach for adaptation on downstream tasks. Full-tuning the entire model often necessitates substantial GPU memories and suffers from the slow training process. Consequently, recent studies in transfer learning have concentrated on optimizing pre-trained models by selecting a limited subset of parameters or introducing extra lightweight parameters. In particular, the prompt-based algorithms (Radford et al., 2018; Brown et al., 2020) advocate for the incorporation of extra trainable tokens to guide the behavior of language models. While this tuning method has also found applications in vision-related downstream tasks (Jia et al., 2022), its drawback lies in the significant drop in accuracy after increasing the prompt number to specific values, as demonstrated in (Chen et al., 2022). The LoRA algorithm (Hu et al., 2021), inspired by the studies of Adapter (Houlsby et al., 2019; Karimi Mahabadi et al., 2021), offers an alternative solution by injecting low-rank matrices into the original attention block. Subsequently, the AdaptFormer (Chen et al., 2022) shifts the perturbation to the feed-forward layer and places it in parallel to the original ViT block. This departs from the setting in LoRA, where low-rank matrices are nested within the attention module. Neural architecture search methods have also been utilized in the following studies (Zhang et al., 2022; Chavan et al., 2023), in order to search a PEFT architecture to maximize the downstream performance. Compared with LoRA-like algorithms, these search methods typically require more training time to identify a proper architecture for each specific downstream task.

Control for Machine Learning Control theory (Franklin et al., 2002; Ogata, 2010) focuses on the analysis and design functions to regulate the behavior of dynamical systems. A cohort of recent studies have utilized machine learning methods to solve the classical control problems, such as the stability analysis (Chang et al., 2019; Dai et al., 2021). But using control to solve practical machine learning problems remains a relatively less explored area. In particular, the early works (E, 2017; Haber & Ruthotto, 2017; Chang et al., 2018) consider the machine learning process, especially learning with ResNet (He et al., 2016), as a function approximation via a control system. Following this understanding, a series of studies (Li et al., 2017; Li & Hao, 2018; Zhang et al., 2019; Kerimkulov et al., 2021) has been working on providing an optimal control viewpoint in the development and understanding of optimization methods for deep learning tasks. In addition to this optimal control view, the controllablity analysis (Ogata, 2010) defines the ability to match arbitrary input and target states, with certain admissible manipulations. It leads to a connection to the classical studies on the expressive ability of continuous-time neural networks (Raghu et al., 2017; Lu et al., 2017). As such, some universal approximation results of deep ResNets have been established based on the controllability analysis (Cheng et al., 2023; Ruiz-Balet & Zuazua, 2023; Li et al., 2022; Cuchiero et al., 2020).

In particular, the controllability analysis of transfer learning with PEFT algorithms has not been explored in the existing literature.

3. A Control Formulation of PEFT Algorithms

We present a control-oriented view for parameter-efficient fine-tuning algorithms in this part, followed by discussions on the principles guiding practical controller design.

3.1. Preliminary and Notations

We begin by revisiting the widely-used Vision Transformer (ViT). Given an image $x_0 \in \mathbb{R}^{C\times H\times W}$, the ViT model first splits and embeds the sample image into a series of visual tokens $x_0' \in \mathbb{R}^{m\times d}$, where m denotes the number of tokens and d refers to the length of each token. Subsequently, an additional class-token $x^{\text{cls}} \in \mathbb{R}^{1\times d}$ is concatenated with these tokens, followed by the addition a positional embedding into each token to form $x_1 \in \mathbb{R}^{(m+1)\times d}$.

These visual tokens are then fed into a set of transformer layers, with the t-th block defined as:

$$x_{t+\frac{1}{2}} = \text{MHSA}(\text{LN}(x_t)) + x_t, \tag{1}$$

$$x_{t+1} = \text{FFN}(\text{LN}(x_{t+\frac{1}{2}})) + x_{t+\frac{1}{2}}, \tag{2}$$

for $t \in [1, \cdots, T-1]$. Here MHSA, FFN and LN denote the multi-head self-attention, feed-forward network and layer normalization, respectively.

The encoded class-token x_T^{cls} will go through a linear layer to conduct the final prediction.

3.2. Dynamics of Controlled ViT Systems

To leverage a pre-trained Vision Transformer (ViT) for downstream tasks without incurring the computational burden associated with full-tuning, a suite of parameter-efficient fine-tuning (PEFT) algorithms has been developed in prior research. In broad terms, these studies concentrate on training specific parts of the original network or incorporating additional lightweight parameters, thereby customizing the pre-trained model for targeted downstream tasks.

In particular, the LoRA algorithm (Hu et al., 2021) endeavors to find a sequence of functions $\{g_t\}$ to be applied to the ViT blocks:

$$x_{t+1} = f_t\left(x_t, \theta_t, g_t(x_t, u_t)\right).$$

Here g_t can be construed as a *control function* that takes the original x_t as input and contains some new parameters u_t. The purpose of this function is to introduce perturbations to the original attention parameter θ_t, which remains constant throughout the learning process.

For efficiency concerns, LoRA employs the linear function $g_t(x_t, u_t) = x_t u_t$, and imposes the constraint that the weight matrix u_t possesses the low-rank property:

$$u_t = A_t B_t, \quad A_t \in \mathbb{R}^{d\times d'}, B_t \in \mathbb{R}^{d'\times d},\ d' \ll d.$$

From a control perspective, the controls within LoRA are integrated into the attention blocks, and analyzing such controls tends to be non-trivial. In contrast, the subsequent AdaptFormer (Chen et al., 2021) relocates the low-rank matrices to the Feed-Forward Network (FFN) layer, positioning them in parallel with the original ViT block. The controlled dynamics can be expressed as:

$$x_{t+1} = f_t(x_t, \theta_t) + g_t(x_t, u_t).$$

A notable advantage of this approach is that the control function g_t is no longer embedded within the original f_t. This decoupling of the control module significantly simplifies the control analysis. As such, we shall adhere to this additive formulation throughout the paper, but consider more general controller designs for g_t.

The overall goal is to design and optimize the parameters of g_t such that the terminal loss on downstream tasks could be minimized:

$$\min_{\{u_t, \theta_T\}} \ \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(x_{\text{Pred},i}, y_i) \tag{3}$$

$$\text{s.t.} \quad x_{t+1,i} = f_t(x_{t,i}, \theta_t) + g_t(x_{t,i}, u_t),\ t \in [1, \cdots, T-1]$$

$$x_{\text{Pred},i} = f_T(x_{T,i}^{\text{cls}}, \theta_T)$$

3.3. Principles of Controller Design

In general, the control function g_t can take arbitrary forms, linear or nonlinear. But for practical implementations, design of such a control module should adhere to two principles: efficiency and controllability.

For efficiency considerations, the parameter number of the control module should be significantly smaller than the original block. This ensures the introduction of extra blocks does not offset our gains from freezing the original ViT blocks. In practice, this reduces the consumption of GPU memories and often accelerates the whole training process.

On the other hand, the controllability defines whether our control blocks possess the capability to steer the ViT system to the desired states.This condition becomes more challenging when operating with constraints imposed by limited parameters.

Due to its inherent low ranks, the AdaptFormer algorithm naturally functions as a parameter-efficient method during the downstream adaptation process. This characteristic enables effective adaptation to specific tasks without the need to tune an excessive number of parameters. But the question remains whether these controlling blocks possess sufficient controllability to steer the ViT system to the desired states.

4. Controllability Analysis

In this section, we present the controllability analysis for the fine-tuning of pre-trained, multi-layer models within a continuous-time context. This analysis aims to provide mathematical perspectives in designing effective controllers for PEFT algorithms.

4.1. A Sufficient Condition for Controllability

We begin by considering a general multi-layer pre-trained model incorporating skip connections between layers (He et al., 2016). The model's dynamics are described as follows:

$$x_{t+1} = x_t + h_t(x_t),\ t = 0, \cdots, T-1, x_t \in \mathbb{R}^D \quad (4)$$

where x_0 is the input, x_T is the output of the model, and h_t denotes the map represented by the t-th layer of the model. Since the parameters of the original model are frozen in the tuning process, we simply use $h_t(x)$ to represent the original dynamics. Viewing the layer index t as a temporal variable transforms our model into a continuous-time analogue, as explored in prior studies (E, 2017; Haber & Ruthotto, 2017):

$$\dot{x}(s) = h(x(s), s), s \in [0, S], \quad (5)$$

where the time s is the continuous analogue of the layer index t. Let $\varphi : x(0) \to x(S)$ denote the input-output relation of dynamics (5). In this framework, we consider the effect of introducing a small scale control function to perturb the model dynamics:

$$\dot{\tilde{x}}(s) = h(\tilde{x}(s), s) + \varepsilon g(\tilde{x}(s), u(s)), \tilde{x}(0) = x(0) \quad (6)$$

where $g : \mathcal{U} \times \mathbb{R}^D \to \mathbb{R}^D$ is the controller with the control parameter $u \in \mathcal{U}$, and $\varepsilon > 0$ is a scaling factor. In the following discussion, we will assume $\mathcal{U}$ to be a compact set. Notice that compared to the general case in (3) where the controller g_t can have different structures across layers, here we are considering the special case where the controller g keeps the same structures, but with different control parameters u across layers. This resembles the setting adopted in current PEFT algorithms.

Let $\varphi_{\varepsilon,u} : \tilde{x}(0) \to \tilde{x}(S)$ denote the perturbed input-output map, which is effectively the feature map of the perturbed model in the deep transformer case. Ideally, we hope a good controller should enable us to adjust the feature map of the perturbed model across a specific dataset. Specifically, for a given set of data samples $X = \{x_i\}_{i=1}^N \subset \mathbb{R}^D$, we expect a good controller to be able to perturb the effect of φ over X arbitrarily, at least on a small scale. This leads us to the concept of local controllability that we now define:

Definition 4.1. We say the system (5) with controller $g(u, x)$ is locally controllable over a dataset $X = \{x_i\}_{i=1}^N \subset \mathbb{R}^D$, if there exists $\varepsilon > 0$ such that the set

$$\{(\varphi_{\varepsilon,u}(x_1), \cdots, \varphi_{\varepsilon,u}(x_N)) \mid u \in L^\infty([0, S], \mathcal{U})\}$$

is an open neighborhood of $(\varphi(x_1), \cdots, \varphi(x_N))$ in $(\mathbb{R}^D)^N$.

Intuitively, suppose the feature map for the downstream task differs from the pre-trained model in a small scale, then a controller meeting Definition 4.1 enables the perturbed feature map to generate the required output features over X, regardless of the distribution of perturbation over the data points in X.

Assume that there is a partition $0 = s_0 < \cdots < s_L = S$ of $[0, S]$, such that $h(x, s)$ is C^2 on each $[s_i, s_{i+1}] \times \Omega$, where Ω is the domain of data. Then, the following theorem gives a sufficient condition on g for the local controllability over X to hold:

Theorem 4.2. *Assume that $g(u, x)$ is locally Lipschitz continuous in both u and x. Also, assume that $g(0, x) \equiv 0$, and for any u and x, there exists v such that $g(v, x) = -g(u, x)$(image set of g is symmetric). For given dataset $X = \{x_i\}_{i=1}^N \subset \mathbb{R}^D$, suppose that the set*

$$\{(g(x_1(s), u), \cdots, g(x_N(s), u) \in (\mathbb{R}^D)^N) \mid u \in \mathcal{U}\} \quad (7)$$

spans $(\mathbb{R}^D)^N$ for each $s \in [0, S]$, where $x_i(s)$ denotes the state of the original dynamics(5) at time s with initial value x_i. Then, the original system with controller g is locally controllable over X.

Proof Idea. Set $\tilde{x}_i(s) = x_i(s) + \varepsilon z_i(s)$, then an asymptotic analysis gives that there exists a uniform constant $C > 0$, such that the solution $\tilde{z}_i(s)$ of

$$\dot{\tilde{z}}_i(s) = \nabla_x^\top h(s, x_i(s)) \cdot \tilde{z}(s) + g(u(s), x_i(s)),\ \tilde{z}_i(0) = 0 \quad (8)$$

satisfies $\|z_i(s) - \tilde{z}_i(s)\| \le C\varepsilon$ for all initial value x_i, $s \in [0, S]$ and $u(\cdot) \in L^\infty([0, S], \mathcal{U})$. By the theory of linear ODE, the solution of (8) at the terminal time T is given by

$$\tilde{z}_i(S) = \mu_i(S) \int_0^S \mu_i^{-1}(s) g(u(s), x(s)) \mathrm{d}s, \quad (9)$$

where $\mu_i(t) \in \mathbb{R}^{d \times d}$ is a fundamental solution matrix of the homogeneous equation of (8), i.e.

$$\dot{\mu}_i(s) = \nabla_x^\top h(x_i(s), s) \cdot \mu_i(s), \quad \mu(0) = I_d. \quad (10)$$

Therefore, $\tilde{z}_i(S)$ depends linearly on $g(u(s), x_i(s))$ for each $s \in [0, S]$. We then split the time interval $[0, S]$ into N small subintervels $[s_j, s_{j+1}], i = 0, \cdots, N$. When N is large enough, for any given perturbation vector $V \in (\mathbb{R}^D)^N$, since $\mu(s)$ is continuous in s and g satisfies the condition in the theorem, one can choose piece-wise constant u such that the integration of

$$(\mu(S)\mu^{-1}(s) g(u(s), x_1(s)), \cdots, \mu(S)\mu^{-1}(s) g(u(s), x_N(s))) \quad (11)$$

over each $[s_j, s_{j+1}]$ is along the same direction with V_j. This yields that $(\tilde{z}_1(S), \cdots, \tilde{z}_i(S))$ can take any value in a some neighborhood of the origin in $(\mathbb{R}^D)^N$. Since $z_i(S)$ differs from $\tilde{z}_i(S)$ with only $\mathcal{O}(\varepsilon)$, we conclude that the local controllability holds for small ε. □

Remark 4.3. The full proof of Theorem 4.2 is given in the appendix. The insight of Theorem 4.2 is actually straightforward: suppose the controller allows all the directions over $(\mathbb{R}^D)^N$ to be admissible for perturbation at each time horizon, then one has the freedom to perturb the final outputs of the dynamics arbitrarily in a small scale.

4.2. Insights on the benefits of nonlinear controller

While the theorem provides just a sufficient condition for local controllability in a continuous-time setting, it yields valuable insights into the controller design. Observations drawn from the condition in Theorem 4.2 reveal that the linear controller is unable to satisfy the condition in Theorem 4.2 when the dataset X is large. In particular, we have the following Proposition.

Proposition 4.4. *Suppose $g(u, x) = A(u)x + B(u)$ which is linear in x, then the condition in Theorem 4.2 cannot hold when $N > D + 1$.*

A recent study (Luo et al., 2023) demonstrates that it is empirically safe to eliminate the ReLU function from AdaptFormer, resulting in a linear controller. But the above Proposition indicates that the expressive ability of the linear controller is highly related to the original dynamics, where it can be possibly compromised if the original dynamics exhibit near-linear behavior across a specific dataset. On the other hand, nonlinear controllers can have the potential to satisfies the condition in Theorem 4.2 for more general datasets X, which is an implication of a strong controllability. This observation then motivates us to consider nonlinear controllers for PEFT algorithms.

Moreover, in the case when the original dynamics function $h(x, s)$ is linear in x, the effect of a linear controller does not add any nonlinear characteristics to the original dynamics. Consequently, the controllability as defined in Definition 4.1 cannot be achieved.

5. Nonlinear Controller Design

Motivated by the above insights suggesting potential benefits of nonlinear controllers, we now turn to the design of a practical nonlinear controller for ViT.

5.1. Cross-Patch Attention is What You Need

Linear control nevertheless offers a straightforward perturbation mechanism on the frozen ViT blocks and are often simple to design and analyze. However, as explored in Section 4.2, the linear controller alone cannot ensure robust controllability for general models; its effectiveness depends on the complexity and nonlinearity inherent in the original dynamics.

Nonlinear control, on the other hand, holds the promise of employing a more complex perturbation mechanism, but the form of such a nonlinear control should be meticulously designed. As an example, the pioneering work AdaptFormer attempts to incorporate nonlinearities by considering $g_t = \sigma(x_t A_t)B_t$, where σ represents an activation function such as ReLU. But subsequent research demonstrates (Luo et al., 2023) that such an activation function has minimal effects in practice, and the performance of AdaptFormer closely resembles that of its linear counterpart.

To get more insights on the design of nonlinearity, let us delve into the original ViT system. In the dynamics of ViT, the state $x = (x^1, \cdots, x^m)^T \in \mathbb{R}^{m\times d}$ is consisting of a sequence of tokens. Note each ViT block comprises two consecutive components: MHSA and FFN. In particular, each head within the MHSA block conducts an attention mechanism via:

$$\text{Attn}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V. \tag{12}$$

A key insight for this attention module is that the tokenized patches are intermingled to generate the new tokens. The following FFN block merely offers a patch-independent linear transformation. Yet, such a "cross-patch attention" (CPA) mechanism is missing in the design of algorithms like AdaptFormer: each tokenized patch is projected downward and upward through linear transformations, with no inclusion of a token mixture process. Such a token-wise controller design can limit the controllability over general models and data, as shown in the following proposition.

Proposition 5.1. *Suppose the controller has the form:*

$$g(u, x) = (\tilde{g}(u, x^1), \cdots, \tilde{g}(u, x^m))^T \tag{13}$$

which is token-wise applied to x, then the condition in Theorem 4.2 cannot hold when there exists some patch $x_{i_1}^{j_1}$ and $x_{i_2}^{j_2}$ of x_{i_1} and x_{j_1} in X, such that $x_{i_1}^{j_1} = x_{i_2}^{j_2}$.

In addition, if the original dynamics $h(s, x)$ are also applied token-wise, there exists cases where the controllability defined in Definition 4.1 is compromised. This observation indicates the necessities of cross-patch information in tuning states that sharing common patches in the original dynamics.

5.2. Nonlinear Controller Design

The above Proposition motivates us to devise a nonlinear control mechanism well-suited for the transformer. In particular, the key step in designing an effective nonlinear controller lies in how to introduce the CPA mechanism, while

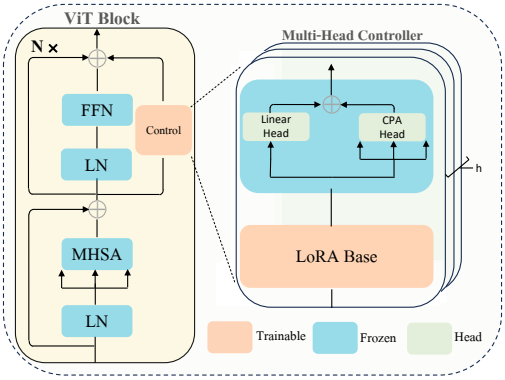

Figure 1. ViT with nonlinear controls.

at the same time minimizing the introduction of additional parameters.

This paper explores the utilization of the LoRA matrices as the base, and proposes a series of heads with minimal parameters. The objective is to ensure the intermingling of tokens from different patches through the implementation of these heads.

In particular, we explore both linear and nonlinear heads, as depicted in Figure 1. For linear heads, a straightforward approach is to direct the controller to output the LoRA results. Regarding nonlinear heads, we examine the following mechanism:

$$x^{p(j')} = \text{CPA}(x) = \sum_j \frac{\exp\langle x^{p(i)}, x^{p(j)}\rangle}{\sum_m \exp\langle x^{p(i)}, x^{p(m)}\rangle} x^{p(j)}, \tag{14}$$

where $x^{p(j)}$ denotes the j-th patch of x. Note in the above CPA design, we refrain from introducing a linear transformation for x to obtain Q, K, V, and we also omit the scaling factor $\sqrt{d}$. This deliberate omission allows the attention to be performed in a parameter-free manner, with only matrix multiplication for x being necessary to compute attention.

The nonlinear output is subsequently combined with the linear output to generate the overall control, resulting in the following control formulation:

$$\text{Control}(x) = \text{LoRA}(x) + \text{CPA}\left(\text{LoRA}(x)\right) \tag{15}$$

And dynamics of controlled-ViT can be depicted as:

$$x_{t+\frac{1}{2}} = \text{MHSA}(\text{LN}(x_t)) + x_t, \tag{16}$$

$$x_{t+1} = \text{FFN}(\text{LN}(x_{t+\frac{1}{2}})) + \text{Control}(x_{t+\frac{1}{2}}) + x_{t+\frac{1}{2}}. \tag{17}$$

Let us make a few comments to the control design. The nonlinear head (14) within the controller has no parameters, ensuring that the total number of parameters in our approach remains identical to the previous work. From a control perspective, incorporating such a head introduces both the nonlinearities and cross-patch information to the controller.

From the machine learning aspect, such a parameter-free head allows the patches to be mixed up to generate the new tokens, akin to the standard attention mechanism. Note the control is applied in parallel to the FFN layer in the above formulation, following the pioneering work (Chen et al., 2022). Alternatively, it could be applied to the entire ViT block, and we observe only very minimal differences in practical scenarios.

Finally, while proving this controller's effectiveness in general cases is challenging, we provide numerical demonstrations that illustrate its satisfaction of the sufficient condition on ViT examples, as later discussed in Section 6.2.

5.3. Multi-Head Controller

One may consider the attention shares the weight of LoRA matrices in the above formulation (14), namely:

$$Q = K = V = A_t B_t. \tag{18}$$

This allows the controller to utilize the existing parameters of LoRA and refrains from introducing extra parameters.

A simple extension of this parameter-sharing mechanism is to increase the head numbers, in order to boost the complexity of controller. In Figure 1, we consider such a scenario by utilizing a multi-head controller. Note such a setting introduces extra parameters and we limit the head number to 2 in this paper for efficiency concerns.

6. Experiment

In this part, we evaluate the effectiveness of the nonlinear controllers by conducting a series of experiments on vision datasets.

6.1. Preliminary

Experimental Settings. For fair comparison, we *mirror* the experimental settings in AdaptFormer (Chen et al., 2022). This involves utilizing the same pretrained Vision Transformer (ViT) backbone, choosing identical downstream tasks, and configuring parameters based on the specifications provided in their original study.

Competing Algorithms. The proposed Attention-augmented Nonlinear Control Algorithm, denoted as ANC, is compared with a few commonly used 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) (Jia et al., 2022): concatenating a set of trainable tokens with existing image tokens; (4) Low-Rank Adaptation (LoRA) (Hu et al., 2021): injecting trainable low-rank matrices to W_Q and W_V; (5) AdaptFormer (Chen et al., 2022): a vision-specific LoRA algorithm by perturbing the FFN layer with (almost) linear controls.

6.2. A Toy Example On the Controllability

We commence with a numerical verification of the condition outlined in Theorem 4.2 through a small-size example. In particular, we consider a scenario wherein the original model is a randomly initialized 10-layer ViT model. The state x encompasses 4 tokens in dimension 5. Subsequently, we introduce our controller defined as follows:

$$g(u, x) := Ax + b + \text{CPA}(Ax + b),$$

and proceed to numerically evaluate the conditions outlined in Theorem 4.2 at each layer of the model.

We randomly generate 20 tokens $x^1, \cdots, x^{20}$ as the input dataset X, and 4000 samples $\{u_j = (A_j, b_j) \mid j = 1, \cdots, 4000\}$, as the control parameters. Subsequently, at a fixed layer $t \in \{1, \cdots, 10\}$, we compute the vectors

$$(g(u_j, x_t^1), \cdots, g(u_j, x_t^{20}))$$

for each $j = 1, \cdots, 4000$. The generated vectors are normalized and stacked into a 4000×400 matrix. Next, we compute the singular values and contrast them with the singular values of the matrix obtained using the linear controller. The comparison of the 5-th transformer block is presented in Figure 2.

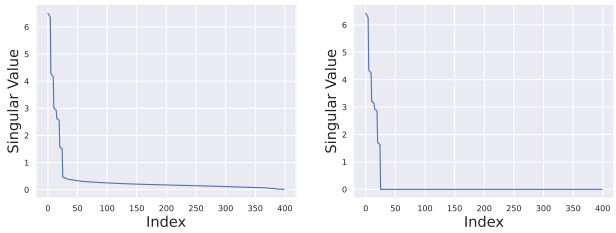

Figure 2. Comparison between the singular value of the matrix obtained by our controller (left) and the linear controller (right) at the 5-th transformer block.

At each block in the model, all singular values with an index greater than 30 become zero in the case of the linear controller, indicating that the condition in Theorem 4.2 cannot be satisfied. In contrast, for our controller, the singular values exhibit a much slower decay, with a minimum value around 0.04. This signifies that the condition in Theorem 4.2 holds for our controller.

6.3. Experiments on Vision Benchmarks

With the above observation, we now proceed to validate our approach on various vision benchmarks, including CIFAR-100 (Krizhevsky et al., 2009), SVHN (Netzer et al., 2011) and Food-101 (Bossard et al., 2014).

Table 1 reports the performance of all algorithms, with the same pre-trained ViT backbone. Notably, the Full-Tuning algorithm consistently achieves the highest levels of test accuracy across all three datasets, establishing itself as a practical upper limit for LoRA-like algorithms. In contrast, the linear-probing algorithm experiences a considerable decline in accuracy ranging from 18.07% to 30.76% across these datasets, indicating that the mere insertion of a linear layer may not yield satisfactory results. This observation also highlights the necessities of perturbing the lower levels of a ViT model to enhance performance.

The PEFT algorithms on the other hand surpass the linear-probing algorithm by a large margin. In particular, we compare the LoRA, AdaptFormer and ANC algorithm with the same number of parameters in Table 1. Note the LoRA algorithm needs to set a low-rank perturbation to both W_Q and W_V, hence its rank would be half of the other two algorithms with equivalent parameter counts. The findings reveal that the Attention-augmented Nonlinear Control algorithm (ANC) consistently outperforms all competing PEFT algorithms across diverse datasets and for all rank configurations. As an illustration, on the Food-101 dataset, the performance demonstrates a notable improvement from 85.70% of LoRA-16 to 88.06% of ANC-32, achieved without introducing any additional parameters. The gap to Full-Tuning could be decreased from 4.39% to 2.03% in this case, where PEFT algorithms only need to tune a small fraction (0.78%) of parameters.

Table 1. Comparison of algorithm performance. We reuse the data reported in AdaptFormer and present single-run results, while repeated experiments for ANC are provided in Appendix B.3. [†] lr for the full-tuning algorithm is decreased by 0.1 to maximize performance of full-tuning on the CIFAR-100 dataset.

Algorithm	# Params(M)	CIFAR-100	SVHN	Food-101
Full-Tuning	86.04	87.90[†]	97.67	90.09
Linear-Probing	0.07	69.83	66.91	69.74
VPT	0.08	82.44	94.02	82.98
LoRA-16	0.67	85.31	96.29	85.70
AdaptFormer-32	0.67	85.42	96.45	86.21
ANC-32	0.67	**86.69**	**96.94**	**88.06**
LoRA-32	1.26	85.42	96.42	86.09
AdaptFormer-64	1.26	85.90	96.89	87.61
ANC-64	1.26	**87.06**	**97.03**	**88.33**
LoRA-64	2.44	85.88	96.58	86.42
AdaptFormer-128	2.44	86.12	96.92	87.78
ANC-128	2.44	**87.17**	**97.11**	**88.50**

By examining the training curves in Figure 3, we notice the ANC-64 algorithm consistently obtains the lowest training loss during the learning process. The terminal training loss

is 19.23% smaller than AdaptFormer-64. Its superior fitting capability is reflected in a relatively higher test accuracy, as demonstrated in Table 1, notwithstanding the models sharing identical parameter counts. This observed phenomenon persists across all datasets and rank configurations, suggesting the superior approximation ability of this nonlinear control in practical applications.

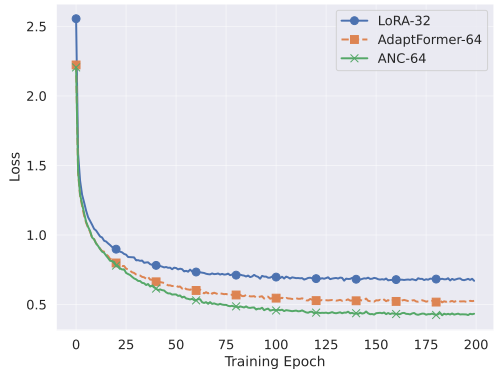

Figure 3. Comparison of training loss for different algorithms on the CIFAR-100 dataset. Similar results on SVHN and Food-101 are available in Appendix B.2.

6.4. Multi-Head Controller Experiment

The incorporation of a nonlinear transformation in the ANC algorithm allows the perturbations to be performed in a more complex way, where different patches could be mixed together through a parameter-free attention mechanism. But the reuse of A_tB_t as Q, K, V nevertheless limits its complexity. To address this limitation, we may increase the head numbers by adopting a multi-head controller, as illustrated in Figure 1. Note such an approach is no longer parameter-free, and we generally need to increase the number of training variables to boost the complexity.

Table 2. Algorithm performance of multi-head controllers. 2H denotes the 2-Head controller.

Algorithm	CIFAR-100	SVHN	Food-101
ANC-32	86.69	96.94	88.06
ANC-32-2H	87.14 (+0.45)	97.03 (+0.09)	88.43 (+0.37)
ANC-64	87.06	97.03	88.33
ANC-64-2H	87.35 (+0.29)	97.12 (+0.09)	88.65 (+0.32)
ANC-128	87.17	97.11	88.50
ANC-128-2H	87.52 (+0.35)	97.18 (+0.07)	89.01 (+0.51)

Table 2 presents the performance of utilizing such a multi-head control strategy. The results reveal that increasing the head number can further boost the overall performance across all datasets and ranks. Notably, this control strategy enables a smaller performance gap to the full-tuning algorithm. For instance, ANC-128-2H attains a test accuracy level of 87.52%, while the full-tuning algorithm achieves 87.90% on the CIFAR-100 dataset. It is noteworthy that the former algorithm only needs to tune 4.82% of the whole parameters, whereas the latter requires full-tuning all 86.04 million trainable variables.

Moreover, by comparing algorithms with the same number of parameters (e.g., ANC-32-2H and ANC-64), it becomes evident that augmenting nonlinearity exhibits a marginally superior performance compared to increasing the rank. Consequently, elevating the rank is no longer the exclusive avenue for enhancing the performance of PEFT algorithms.

6.5. Ablation Studies on Nonlinearity

This paper explores the architectural design by leveraging LoRA matrices as the foundation, supplemented with the incorporation of a Cross-Patch Attention (CPA) head to introduce nonlinearity. This departure from prevailing methodologies, which typically introduce nonlinearity through activation functions between A_t and B_t. We highlight the necessities of token intermingling by contrasting with the following algorithms: (1) the standard AdaptFormer utilizing a ReLU function; (2) a purely linear control method achieved by excluding the ReLU function; (3) a nonlinear control by injecting a sigmoid function into AdaptFormer.

Table 3. Ablation study on the forms of nonlinearity

Algorithm	CIFAR-100	SVHN	Food-101
ANC-64	**87.06**	**97.03**	**88.33**
AdaptFormer-64	85.90	96.89	87.61
Linear-64	86.01	96.85	87.68
AdaptFormer-64-Sigmoid	84.61	95.99	85.80

Table 3 indicates that the incorporation of the nonlinear ReLU function yields marginal impact on the final performance of AdaptFormer, in comparison with its linear counterpart. Moreover, the incorporation of a sigmoid function leads to performance declines across all datasets, notably resulting in a significant decrease of 1.4% on the CIFAR-100 dataset. Consequently, the form of nonlinearity has to be meticulously designed, in order to outperform the pure linear controls. The attention-augmented algorithm consistently outperforms both linear controls and nonlinear controls employing activation functions. Note the rank is set to 64 in Table 3, but the observed phenomenon persists across all rank configurations.

7. Conclusion

This paper bridges the controllability analysis with recent investigations into PEFT algorithms. Specifically, we recast the LoRA-like algorithms as a control problem and conduct a comprehensive analysis of the controllability of low-rank modules, thereby establishing sufficient conditions for downstream controls. The controller modules are further redesigned by introducing nonlinearities through a parameter-free attention mechanism, enabling token inter-

mingling within the controllers. Empirical results demonstrate that this approach outperforms the existing LoRA-like algorithms across all evaluated datasets and rank configurations, without introducing additional parameters.

Impact Statement

This paper aims to bridge parameter-efficient algorithms with control theory. There are minor potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

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

References

Aghajanyan, A., Zettlemoyer, L., and Gupta, S. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

Bapna, A., Arivazhagan, N., and Firat, O. Simple, scalable adaptation for neural machine translation. *arXiv preprint arXiv:1909.08478*, 2019.

Bishop, R. C. D. R. H. *Modern control systems*. 2011.

Bossard, L., Guillaumin, M., and 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.

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.

Chang, B., Meng, L., Haber, E., Ruthotto, L., Begert, D., and Holtham, E. Reversible architectures for arbitrarily deep residual neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Chang, Y.-C., Roohi, N., and Gao, S. Neural lyapunov control. *Advances in neural information processing systems*, 32, 2019.

Chavan, A., Liu, Z., Gupta, D., Xing, E., and Shen, Z. One-for-all: Generalized lora for parameter-efficient fine-tuning. *arXiv preprint arXiv:2306.07967*, 2023.

Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., and Luo, P. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022.

Chen, T., Zhang, Z., Ouyang, X., Liu, Z., Shen, Z., and Wang, Z. " bnn-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.

Cheng, J., Li, Q., Lin, T., and Shen, Z. Interpolation, approximation and controllability of deep neural networks. *arXiv preprint arXiv:2309.06015*, 2023.

Cuchiero, C., Larsson, M., and Teichmann, J. Deep neural networks, generic universal interpolation, and controlled odes. *SIAM Journal on Mathematics of Data Science*, 2 (3):901–919, 2020.

Dai, H., Landry, B., Yang, L., Pavone, M., and Tedrake, R. Lyapunov-stable neural-network control. *arXiv preprint arXiv:2109.14152*, 2021.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

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.

E, W. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 1 (5):1–11, 2017.

Franklin, G. F., Powell, J. D., Emami-Naeini, A., and Powell, J. D. *Feedback control of dynamic systems*, volume 4. Prentice hall Upper Saddle River, 2002.

Haber, E. and Ruthotto, L. Stable architectures for deep neural networks. *Inverse problems*, 34(1):014004, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Jia, M., Tang, L., Chen, B.-C., Cardie, C., Belongie, S., Hariharan, B., and Lim, S.-N. Visual prompt tuning. In *European Conference on Computer Vision*, pp. 709–727. Springer, 2022.

Karimi Mahabadi, R., Henderson, J., and Ruder, S. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34: 1022–1035, 2021.

Kerimkulov, B., Šiška, D., and Szpruch, L. A modified msa for stochastic control problems. *Applied Mathematics & Optimization*, pp. 1–20, 2021.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kwakernaak, H. and Sivan, R. *Linear optimal control systems*, volume 1. Wiley-interscience New York, 1972.

Lewis, F. L., Dawson, D. M., and Abdallah, C. T. *Robot manipulator control: theory and practice*. CRC Press, 2003.

Li, Q. and 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.

Li, Q., Chen, L., Tai, C., et al. Maximum principle based algorithms for deep learning. *arXiv preprint arXiv:1710.09513*, 2017.

Li, Q., Lin, T., and Shen, Z. Deep learning via dynamical systems: An approximation perspective. *Journal of the European Mathematical Society*, 25(5):1671–1709, 2022.

Lian, D., Zhou, D., Feng, J., and Wang, X. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35:109–123, 2022.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. The expressive power of neural networks: A view from the width. *Advances in neural information processing systems*, 30, 2017.

Luo, G., Huang, M., Zhou, Y., Sun, X., Jiang, G., Wang, Z., and Ji, R. Towards efficient visual adaption via structural re-parameterization. *arXiv preprint arXiv:2302.08106*, 2023.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Ogata, K. *Modern control engineering fifth edition*. 2010.

Pan, S. J. and Yang, Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10): 1345–1359, 2009.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 2018.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pp. 28492–28518. PMLR, 2023.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., and Sohl-Dickstein, J. On the expressive power of deep neural networks. In *international conference on machine learning*, pp. 2847–2854. PMLR, 2017.

Ruiz-Balet, D. and Zuazua, E. Neural ode control for classification, approximation, and transport. *SIAM Review*, 65 (3):735–773, 2023.

Slotine, J.-J. E. and Li, W. On the adaptive control of robot manipulators. *The international journal of robotics research*, 6(3):49–59, 1987.

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pp. 843–852, 2017.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Zhang, D., Zhang, T., Lu, Y., Zhu, Z., and Dong, B. You only propagate once: Accelerating adversarial training via maximal principle. *Advances in Neural Information Processing Systems*, 32, 2019.

Zhang, Y., Zhou, K., and Liu, Z. Neural prompt search. *arXiv preprint arXiv:2206.04673*, 2022.

A. Proof for Theorems and Propositions

A.1. Proof of Theorem 4.2

Set $\tilde{x}_i(s) = x_i(s) + \varepsilon z_i(s)$ and substitute it into the perturbed dynamics (6), we have:

$$\dot{x}_i(s) + \varepsilon \dot{z}_i(s) = h(\tilde{x}_i(s), s) + \varepsilon g(\tilde{x}_i(s), u(s)). \tag{19}$$

Subtracting the original dynamics, and expand with respect to ε gives:

$$\dot{z}_i(s) = \nabla_x^\top h(x_i(s), s) \cdot z_i(s) + g(u(s), x_i(s)) + \delta_i(s),\ z(0) = 0. \tag{20}$$

Since h is piece-wise C^2, g is locally Lipschitz, $\mathcal{U}$ is compact, we deduce that there exists constant C, which depends only on g, h such that $\|\delta_i(s)\| \leq C\varepsilon$ for all x_i and $s \in [0, S]$. Therefore, if we denote $\tilde{z}$ as the solution of the following linear ODE:

$$\dot{z}_i(s) = \nabla_x^\top h(x_i(s), s) \cdot z_i(s) + g(x(s), u(s)),\ z(0) = 0, \tag{21}$$

then the difference between $z(s)$ and $\tilde{z}(s)$ is in $\mathcal{O}(\varepsilon)$. Let $\mu(s) \in \mathbb{R}^{d\times d}$ be a fundamental solution matrix of the homogeneous equation of (8), i.e.

$$\dot{\mu}(s) = \nabla_x^\top h(x(s), s) \cdot \mu(s), \quad \mu(0) = I_d, \tag{22}$$

By the theory of linear ODE, the solution of (8) is given by

$$z(s) = \mu(s) \int_0^t \mu^{-1}(s) g(x(s), u(s)) \mathrm{d}s$$

For the proof of Theorem 4.2, we need the following technical lemma. The lemma states that the spatial average of trajectories can be realized via average in time.

Lemma A.1. *Let $x_k(\cdot) : [0, S] \to \mathbb{R}^D, k = 1, \cdots, x_N$ be the N different trajectories of the original ODE. Suppose $\gamma(x, s) : [0, S] \times \mathbb{R}^D \to \mathbb{R}^D$ is a continuous map satisfies that for any $s \in [0, S]$, there exists $\lambda_1, \cdots, \lambda_m \in \mathbb{R}$ and $u_1, \cdots, u_m \in \mathcal{U}$, such that $|\lambda_1| + \cdots + |\lambda_m| \leq 1$ and*

$$\gamma(x_k(s), s) = \sum_{i=1}^m \lambda_i g(x_k(s), u(s)), \quad \textit{for all } k = 1, \cdots, N.$$

Then, for any $\delta > 0$, there exists a trajectory $u(\cdot) : [0, S] \to \mathcal{U}$ such that

$$\left| \int_0^S \mu_k^{-1}(s) \gamma(x_k(s), s) \mathrm{d}s - \int_0^S \mu_k^{-1}(s) g(x_k(s), u(s)) \mathrm{d}s \right| < \delta, \textit{ for all } k = 1, \cdots, N,$$

where $\mu_k(\cdot)$ denotes the trajectory of (22) where the trajectory $x(\cdot)$ is given by $x_k(\cdot)$.

Proof of lemma. Let $M_1 := \sup_{k,s} \|h(x_k(s), s)\|$, $M_2 := \sup_{k,s,u} \|g(x_k(s), u(s))\|$, $M_3 := \sup_{s,k} \|\mu_k(s)^{-1}\|_2$. Since f is piece-wise C^1, g is continuous, $\mathcal{U}$ is bounded, we have that $M_1, M_2, M_3 < \infty$. Let L_g and L For integer $N > 0$, we split the interval $[0, S]$ into N subintervals $[t_{j-1}, t_j],\ j = 1, \cdots N$, where $t_j = \frac{j}{N} S$. At each t_j, there exists $\lambda_{j,1}, \cdots, \lambda_{j,m_j} \in \mathbb{R}$ and $u_{j,1}, \cdots, u_{j,m_j} \in \mathcal{U}$, such that $|\lambda_{k,1}| + \cdots + |\lambda_{j,m_j}| \leq 1$ and

$$\gamma(x_k(t_j), t_j) = \sum_{i=1}^{m_j} \lambda_{j,i} g(x_k(t_j), u_{j,i}), \quad \text{for all } k = 1, \cdots, N.$$

For convenience, let $\beta_l := \sum_{i=1}^l \lambda_{j,i}$ for $l = 0, \cdots, m_k$, and let $\beta_{m_{k+1}} = 1$. Then, we define

$$u(s) = u_{j,i}, \quad \text{for } s \in [t_{j-1} + \beta_{l-1}(t_j - t_{j-1}), t_{j-1} + \beta_l(t_j - t_{j-1})),\ l = 1, \cdots, m_j,$$

and $u(s) = 0$ for $s \in [t_{j-1} + \beta_{m_j}(t_j - t_{j-1}), t_j]$. Then, we have:

$$\begin{aligned}
&\left|\int_{t_{j-1}}^{t_j} \mu_k^{-1}(s)\gamma(x_k(s), s)\mathrm{d}s - \int_{t_{j-1}}^{t_j} \mu_k^{-1}(s)g(x_k(s), u(s))\mathrm{d}s\right| \\
\leq &\left|\int_{t_{j-1}}^{t_j} \mu_k^{-1}(s)(\gamma(x_k(s), s)\mathrm{d}s - \gamma(x_k(t_{k-1}), t_{k-1}))\mathrm{d}s\right| + \left|\int_{t_{j-1}}^{t_j} \mu_k^{-1}(s)(\gamma(x_k(t_{j-1}), t_{j-1})\mathrm{d}s - g(x_k(s)), u(s))\mathrm{d}s\right| \\
\leq &(t_j - t_{j-1})M_3 \cdot \omega(\gamma, \frac{S}{N}(1 + M_1)) + \left|\int_{t_{j-1}}^{t_j} \mu_k^{-1}(s)(\gamma(x_k(t_{j-1}), t_{j-1})\mathrm{d}s - g(x_k(s), u(s)))\mathrm{d}s\right|
\end{aligned} \tag{23}$$

where $\omega(f, \delta) = \sup\{|f(x) - f(y)| : |x - y| \leq \delta\}$ denotes the modulus of continuity of a function. For the second term in the last line, we have the estimate

$$\begin{aligned}
&\left|\int_{t_{j-1}}^{t_j} \mu_k^{-1}(s)(\gamma(x_k(t_{j-1}), t_{j-1})\mathrm{d}s - g(x_k(s), u(s)))\mathrm{d}s\right| \\
\leq &\sum_{l=1}^{m_k}\sum_{p=1}^{m_k}\left|\lambda_{j,l}\int_{t_{j-1}+\beta_{p-1}(t_j - t_{j-1})}^{t_{j-1}+\beta_p(t_j - t_{j-1})} \mu_k^{-1}(s)(g(x_k(t_{j-1}), u_{j,l}))\mathrm{d}s - \lambda_{j,p}\int_{t_{j-1}+\beta_{l-1}(t_j - t_{j-1})}^{t_{j-1}+\beta_l(t_j - t_{j-1})} \mu_k^{-1}(s)g(x_k(s), u_{j,l})\mathrm{d}s\right| \\
\leq &(t_j - t_{j-1}) \cdot \sum_{l=1}^{m_k}\sum_{p=1}^{m_k}\lambda_{j,l}\lambda_{j,p}(M_3\omega(g, \frac{M_2 S}{N}) + M_2\omega(\mu_k^{-1}, \frac{S}{N})) \\
= &(t_j - t_{j-1}) \cdot (M_3\omega(g, \frac{M_2 S}{N}) + M_2\omega(\mu_k^{-1}, \frac{S}{N}))
\end{aligned} \tag{24}$$

Combining (23) and (24), we have

$$|\int_0^S \mu_k^{-1}(s)\gamma(x_k(s), s)\mathrm{d}s - \int_0^S \mu_k^{-1}(s)g(x_k(s), u(s))\mathrm{d}s| \leq S(M_3\omega(\gamma, \frac{S}{N}(1+M_1)) + M_3\omega(g, \frac{M_2 S}{N}) + M_2\omega(\mu_k^{-1}, \frac{S}{N})). \tag{25}$$

Since γ, g and μ_k^{-1} are continuous, the lemma is proved when N goes to infinity. □

Proof of Theorem 4.2 . By the assumption of the theorem, for any $s \in [0, S]$, there exists an $r(s) > 0$ such that the ball with radius $r(s)$

$$B_{r(s)} := \{x \in \mathbb{R}^{Nd} \mid \|x\| < r(s)\}$$

is contained in the set

$$C(s) := \{\sum_{i=1}^{m} \lambda_i \left(g(x_1(s), u_i), g(x_2(s), u_i), \cdots, g(x_N(s), u_i)\right) \mid m \in \mathbb{Z}^+, |\lambda_1| + \cdots + |\lambda_m| \leq 1\}.$$

Therefore, for any vector $V = (v_1, \cdots, v_N) \in \mathbb{R}^{ND}$(where $v_i \in \mathbb{R}^D$) with unit norm, one can construct a continuous function $\gamma(x, s)$ such that

$$(\mu_1(S)\mu_1^{-1}(s)\gamma(x_1(s), s), \mu_2(S)\mu_2^{-1}(s)\gamma(x_2(s), s), \cdots, \mu_N(S)\mu_N^{-1}(s)\gamma(x_N(s), s))$$

is a.e. non-zero for $s \in [0, S]$ and is along the same direction with V. Therefore, the vector

$$\left(\mu_1(S)\int_0^T \mu_1^{-1}(s)\gamma(x_1(s), s)\mathrm{d}s, \cdots, \mu_N(S)\int_0^T \mu_N^{-1}(s)\gamma(x_N(s), s)\mathrm{d}s\right) = cV.$$

for some positive constant c. According to Lemma A.1, this implies that

$$\left(\mu_1(S)\int_0^S \mu_1^{-1}(s)g(x_1(s), u(s),)\mathrm{d}s, \cdots, \mu_N(S)\int_0^S \mu_N^{-1}(s)g(x_N(s), u(s))\mathrm{d}s\right)$$

can be arbitrarily close to cV when $u(\cdot)$ varies. Since V is arbitrary and $z(S) = \mu(S)\int_0^S \mu^{-1}(s)g(x(s), u(s))\mathrm{d}s$ is the leading term of $\varphi_{\varepsilon,u} - \varphi$, we deduce that there exists $\varepsilon > 0$ such that the set

$$\{(\varphi_{\varepsilon,u}(x_1), \cdots, \varphi_{\varepsilon,u}(x_N)) \mid u \in L^\infty([0, S])\}$$

is an open neighborhood of $(\varphi(x_1), \cdots, \varphi(x_N))$. □

A.2. Proof for Proposition 4.4

Proof. When the dataset X is fixed, the expression

$$A(u)x + B(u) = \sum_{1\le i,j\le D} A_{ij}(u)(E_{ij}x) + B_i(u)e_i$$

is a linear combination of no more than $D^2 + D$ fixed vectors, where E_{ij} are the (i, j)-th matrix unit and e_i are the standard basis vectors for $\mathbb{R}^d$. Therefore, the space spanned by (7) is at most in $D^2 + D$ dimension. Then, when $N > D + 1$, $N \times D$ will surpass $D^2 + D$ and the condition in Theorem 4.2 cannot hold. □

A.3. Proof of Proposition 5.1

Proof. For convenience, suppose that the first token x_1^1 and x_2^1 of x_1 and x_2 are the same. Since g is token-wise applied, this implies that the first token of $g(x_1, u)$ and $g(x_2, u)$ will always be the same.

Therefore, the set in (7) will be restricted to a subspace with co-dimension at least d. □

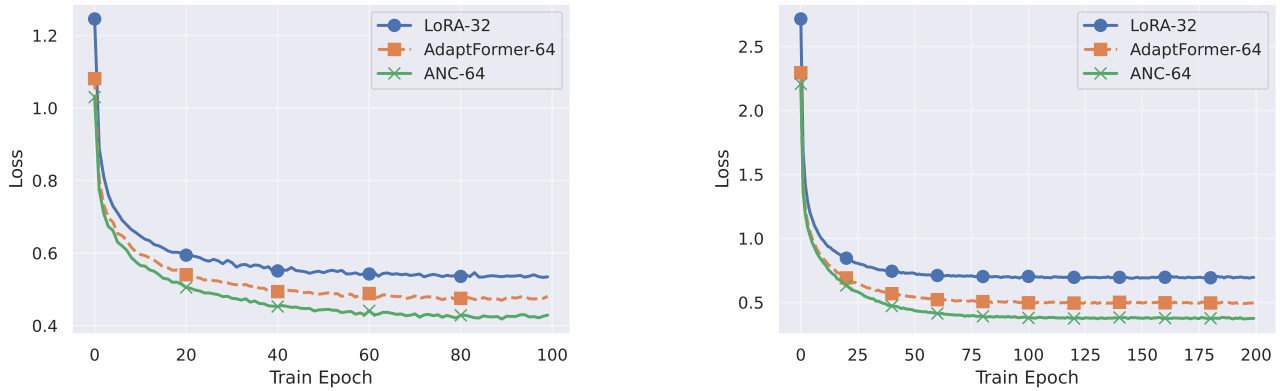

Figure 4. Train loss on the SVHN (left) and Food-101 (right) dataset.

B. Additional Experiments

B.1. Experimental Settings

In general, the experimental configurations adhere to the settings employed in the prior AdaptFormer (Chen et al., 2022) study. A plain Vision Transformer (ViT-Base) model serves as the underlying architecture, pre-trained on the ImageNet-21K dataset (Deng et al., 2009) with MAE (He et al., 2022). The down-projection layer weights in the controls are initialized using Kaiming Norm (He et al., 2015), while the up-projection layer weights are set to 0. Analogously, all biases in the controls are initialized to 0.

The Stochastic Gradient Descent (SGD) algorithm with a momentum of 0.9 is employed for optimizing the controls during the training process. Its batch-size is set to 128 and the learning rate is set to 0.05. All experiments are conducted on the Nvidia-3090.

B.2. Training Loss for SVHN and Food-101 Datasets

We report the training curves on the CIFAR-100 dataset in Figure 3. For completeness, the corresponding curves for the SVHN and Food-101 datasets are depicted in Figure 4. The observed results indicate that the ANC algorithm consistently achieves the lowest training losses on these two datasets.

B.3. Repeated Experiments for ANC

Table 4. Repeated experiments on the ANC algorithm.

Algorithm	CIFAR-100	SVHN	Food-101
ANC-32	86.68 ± 0.03	96.96 ± 0.03	88.08 ± 0.05
ANC-64	87.06 ± 0.02	97.04 ± 0.02	88.33 ± 0.03
ANC-128	87.18 ± 0.03	97.12 ± 0.02	88.50 ± 0.05
ANC-32-2H	87.15 ± 0.03	97.05 ± 0.02	88.42 ± 0.04
ANC-64-2H	87.35 ± 0.03	97.12 ± 0.02	88.66 ± 0.04
ANC-128-2H	87.51 ± 0.07	97.18 ± 0.02	89.03 ± 0.05

We present the algorithm performance on a single-run in Table 1. In particular, we set the seed as 42 whenever possible. As such, the up and down projections in all LoRA-like algorithms are initialized with the same value, so that the only difference lies in the control architecture. This approach effectively mitigates the impact of divergent initializations across different algorithms.

For completeness, we repeat the ANC experiments thrice with different seeds, and report its performance in Table 4. The results demonstrate that the performance of ANC remains generally stable across different seed values, exhibiting minimal variances.