Performance Comparison of Reversible Vision Transformer Models

Haolun Yang
Department of Computer Science
University of Exeter, Exeter, UK

Internal Supervision:
Guoqiang Zhang
Department of Computer Science

***Abstract*—This study explores performance optimization and resource consumption control in Vision Transformer (ViT) models by comparing the performance of ViT-small, Reversible ViT-small, and BDIA ViT-small on the CIFAR-10 and MNIST datasets. The experimental results reveal that the BDIA ViT-small model outperforms the others in accuracy and stability while achieving a good balance in memory usage. Although the Reversible ViT-small model excels in memory efficiency, its accuracy is slightly lower than the standard ViT-small model. These findings provide valuable insights for optimizing ViT models in resource-constrained environments.**

I. Introduction

With the rapid development of computer vision, Vision Transformer (ViT) models have become a research hotspot due to their outstanding performance in tasks such as image classification and object detection. However, as the scale and complexity of these models increase, particularly in practical applications, it has become an urgent challenge to control resource consumption while ensuring effective high performance. This study focuses on finding the best balance between performance and resource consumption in the Visual Transformer models. It explores techniques to improve the performance of the Reversible Visual Transformer to resolve the above contradictions.

In this context, the study of Reversible Neural Networks is of great significance. Reversible Neural Networks can significantly reduce memory consumption by recomputing intermediate activations during backpropagation. However, most existing reversible neural networks require substantial modifications to the original model architecture or are constrained by specific structural designs, which limits their broader applicability. Therefore, achieving efficient reversibility without altering the model architecture has become an important research direction.

This study aims to compare the performance and resource consumption of three Vision Transformer models to explore the optimal model architecture. Specifically, we have selected the following three models for investigation: First, ViT-small, a standard Vision Transformer model designed for small datasets, which has demonstrated efficient performance on such datasets and will serve as the base model for this research, with other reversible structures derived from it [1]. Second, Reversible ViT-small introduces a reversible architecture based on ViT-small to reduce memory usage. Finally, BDIA ViT-small, which further incorporates the Bidirectional Integration Approximation (BDIA) technique. This model treats each Transformer block as an Euler integration approximation of an ordinary differential equation (ODE) and, combined with activation quantization, achieves precise bit-level reversibility, thereby reducing memory consumption. Additionally, the introduction of a random hyperparameter γ during training effectively regularizes the model, improving validation accuracy. Although the initial project proposal considered introducing momentum into the Reversible Vision Transformer to smooth the training process, this idea encountered several compatibility issues during the experiments, which will be discussed in the future outlook section.

In practical applications, the resource consumption of a model directly affects its deployability and universality. With the widespread adoption of deep learning, particularly in resource-constrained environments such as edge devices and mobile devices, the ability to design models that are both efficient and resource-friendly is crucial to the large-scale promotion of the technology. Therefore, this study not only advances our understanding of how to optimize Vision Transformer models but also provides practical guidance for designing deep learning models in resource-constrained environments.

Through this study, we will reveal how the visual Transformer model achieves the best balance between performance and resource consumption under different technologies. The study will not only help develop more efficient model architectures but also provide a strong theoretical basis for applying these technologies to larger datasets and more complex tasks in the future. Future research can further explore how to use these technologies to optimize model performance in different application scenarios, thereby promoting the application of the visual Transformer model in a wider range of fields.

II. Related work

This section provides a detailed overview of the background of Vision Transformer models and reversible neural network models. It discusses the recent developments in Vision Transformers as reported in the literature, as well as the application of reversible techniques in neural networks. Furthermore, it analyzes the key focuses of some significant studies, highlighting the strengths and weaknesses of these models. Additionally, a brief introduction to the algorithms used in the complex models for the comparative experiments is provided.

A. Vision Transformers

With the continuous development of deep learning in the field of computer vision, the Vision Transformer (ViT) model has emerged as a new architecture. This model demonstrates good performance in tasks such as image classification and object detection. The Transformer architecture initially showcased its strong capabilities in natural language processing (NLP) tasks. The BERT model, proposed by Jacob Devlin, Ming-Wei Chang, and their team, introduced a technique that enables joint tuning of context from all layers, achieving bidirectional deep representations of unannotated text. This allowed the pre-trained model to perform various language processing tasks with only fine-tuning of the output layer, achieving top accuracy in up to 11 NLP tasks. In 2020, Dosovitskiy et al. applied this architecture to the visual domain and proposed the ViT model [2]. Their work demonstrated the feasibility of directly applying the self-attention mechanism to process image patches. Specifically, the input image is divided into fixed-size patches, with each patch treated as an input "token" for self-attention computation. In this way, the ViT model captures global image features without relying on convolution operations, showing great potential in complex visual tasks.

ViT achieved significant success in image classification tasks, but its reliance on large-scale datasets and suboptimal performance on smaller datasets have become a focus of researchers. The knowledge distillation technique used in the DeiT model allows ViT to perform well on small datasets even without large-scale pre-training [3]. The DeiT model uses an additional distillation token to help the student model learn from the teacher model. The distillation token, along with image patches and the class token, is input into the Transformer, and the final output is determined by both the classification head and the distillation head.

Touvron et al. revisit the training procedures for Vision Transformers (ViTs) and propose an improved recipe for training these models that builds upon and simplifies previous methods [4]. Their approach addresses the challenges faced by ViTs when trained on midsize datasets like ImageNet-1k, which are critical for many computer vision tasks. The authors introduce a new data augmentation strategy called "3-Augment," which simplifies the augmentation process by using only three transformations: grayscale, solarization, and Gaussian blur. This method proved to be more effective for ViTs compared to other more complex augmentation strategies typically used for convolutional neural networks. Additionally, they emphasize the importance of using a lower training resolution to reduce the discrepancy between training and test data, ultimately leading to better model generalization and reduced overfitting. Their work demonstrates that, with these improvements, ViTs can achieve competitive performance on tasks such as image classification and semantic segmentation, rivaling more recent architectures like Swin Transformers. The findings of this study provide valuable insights into optimizing ViT training, making these models more efficient and accessible for a broader range of applications.

Other studies have shown that as the model depth increases, ViT may encounter the problem of "attention collapse," where the self-attention distributions across different layers become increasingly similar, leading to a decline in the model's representational capacity. To address this issue, Khawar Islam and his team proposed the Re-attention mechanism [5], which enhances inter-layer diversity of attention maps by regenerating them at different layers, thus improving the performance of deep ViT models.

However, the computational complexity and memory requirements of ViT are significantly higher than those of traditional Convolutional Neural Networks (CNNs), making it challenging to apply in resource-constrained devices. To solve this issue, Chen et al. proposed the Visformer model [6], which combines convolution operations with the self-attention mechanism, thereby maintaining the global modeling capability of ViT while reducing computational resource consumption. Similarly, Namuk Park and Songkuk Kim believe that Vision Transformers and CNNs can be complementary. They proposed the AlterNet model, which integrates the multi-head self-attention (MSA) mechanism of Vision Transformers into CNNs [7], replacing the Conv block at the end of each stage with an MSA block, allowing MSA to play a critical role in predictions.

As the application of Transformer architectures in the field of computer vision continues to deepen, researchers are actively exploring ways to further optimize the ViT model architecture to achieve better performance in more application scenarios. Some research directions include improving regularization methods (such as DropPath) to enhance model training stability, as well as exploring more efficient attention mechanisms like the sliding window attention in Swin Transformer to reduce computational costs [8], and improving the applicability of ViT on small datasets through data augmentation and model compression techniques such as Token Merging [9].

B. Reversible neural network models

Firstly, the RevNets (Reversible Residual Networks) model proposed by Gomez et al. in 2017 laid the foundation for reversible neural networks. Residual networks, through the design of residual blocks, allow information to be directly transmitted, thereby mitigating the problems of vanishing and exploding gradients, which makes training deeper neural networks possible. However, as the network depth increases, traditional residual networks require storing the activation values of each layer during backpropagation, leading to increased memory consumption. To address this issue, Gomez et al. proposed RevNets, which reconstruct the input of the previous layer using the activations of the current layer, thereby reducing memory usage during backpropagation [10]. Specifically, RevNets divide the input data into two parts and operate on them using two functions, F and G. In the forward pass, F acts on x_2 and adds to x_1 to obtain y_1, and then G acts on y_1 and adds to x_2 to obtain y_2. During backpropagation, the

input is reconstructed using the equations $x_2 = y_2 - G(y_1)$ and $x_1 = y_1 - F(x_2)$, while the activations y_1, y_2 and their derivatives are used to calculate the inputs x_1, x_2 and their derivatives, eliminating the need to store the activations of the layers. This modular design is independent of the residual functions F and G, making it generalizable and allowing the network to maintain reversibility in the forward pass while reducing memory requirements in backpropagation. Although this method significantly reduces memory usage, it increases computational cost due to the additional operations required to reconstruct the input. Additionally, unlike traditional ResNets, where layers can have larger strides, the stride in the reversible blocks of RevNets must be 1; otherwise, information from intermediate layers will be lost.

Chang et al. in 2018 further expanded the application of reversible networks by viewing residual networks as a discrete form of ordinary differential equations (ODEs) and introducing structures such as Hamiltonian networks, midpoint networks, and leapfrog networks [11]. These structures achieve reversibility and stability in the network by simulating energy conservation in physical systems and numerical integration methods. Particularly, Hamiltonian networks introduce two sets of convolutional kernels K_1 and K_2, where $K(t)$ and its transpose represent convolution and transposed convolution operations, respectively. The network state updates depend on each other and are mathematically reversible. These structures perform well in handling complex data patterns, especially in reducing memory demands while maintaining high accuracy. However, while these methods offer theoretical advantages, in practical applications, special attention must be paid to numerical stability to avoid instability during forward and backward propagation. The authors also introduced midpoint networks, which are based on the midpoint method for numerical integration and used to discretize ODEs. The forward propagation in these networks is defined by $Y_{j+1} = Y_{j-1} + 2hF(Y_j)$, where h is the step size. In the single-layer midpoint network, when $j > 0$, $F(Y)$ is defined as $\sigma((K - K^T)Y + b)$, allowing the network to recover the state $Y(j)$ from the preceding and succeeding states. Although this network is algebraically reversible, using a double-layer midpoint network is recommended for stability. Additionally, the leapfrog network, a special case of the Hamiltonian network, is a reversible neural network structure based on the leapfrog method from numerical analysis.

To address the high memory demand in computing gradients for Neural ODEs, Gholami et al. proposed the ANODE framework, which effectively reduces memory usage by introducing checkpointing techniques and optimized gradient computation strategies. ANODE views the training process of ODEs as an optimization problem and uses Lagrangian functions to constrain the dynamics of ODEs, thereby reducing memory consumption during backpropagation. This method is particularly suitable for training deep networks, as it significantly reduces memory requirements without increasing computational complexity [12]. However, this method also faces the challenge of increased computational cost, especially in extremely memory-constrained environments, where extensive recomputation is required, significantly raising the computational burden.

Momentum is another technique that can be considered for reversible neural networks to smooth the training process. The team led by M. E. Sander introduced the concept of momentum into reversible residual neural networks, proposing the Momentum Residual Network. They defined the function in the residual block as $v_{n+1} = \gamma v_n + (1-\gamma)f(x_n, \theta_n)$ and $x_{n+1} = x_n + v_{n+1}$, where v is the velocity term, γ is the momentum term, and f is the residual mapping function. Using $x_n = x_{n+1} - v_{n+1}$ and $v_n = \frac{1}{\gamma}(v_{n+1} - (1-\gamma)f(x_n, \theta_n))$, the activation of the nth layer can be fully recovered from the $(n+1)$th layer, achieving network reversibility [13]. Like traditional reversible neural networks, the introduction of momentum does not affect the transmission of residuals between layers, and the input of the next layer can still recover the output of the previous layer during backpropagation without storing intermediate activations during training, thus saving memory. By adjusting the momentum parameter γ within the range [0,1], this network structure can transition from approximating a traditional residual network to a symplectic scheme model similar to RevNet and Hamiltonian networks. The Momentum Residual Network can also be interpreted as a second-order ODE in the continuous-time limit, allowing it to learn more complex dynamics more precisely.

The Reversible Vision Transformer model proposed by Karttikeya Mangalam, Haoqi Fan, and their team also provides important guidance for this experiment. The basic principle of this model is to divide the input tensor I into two d-dimensional tensors (I_1, I_2) and then use the transformation $T1$ to form two parts of the tensor (O_1, O_2). The attention module and feedforward module in the Transformer model are designed as functions F and G, respectively, applied to I_1 and I_2, and the computed outputs (O_1, O_2) are added crosswise, i.e., $O_1 = I_1 + G(I_2)$ and $O_2 = I_2 + F(I_1)$. Since the transformations $T1$ and $T2$ allow inverse transformations and the output is differentiable, the combined transformation T is also reversible. Thus, after such a composite transformation, the original input I can still be recovered from the output O, and this structure is reversible and does not introduce additional computational burden due to activation calculations during backpropagation [14].

C. Bidirectional Integration Approximation

Bidirectional Integration Approximation (BDIA) is a novel method recently proposed in the field of diffusion inversion, aiming to address the high computational cost and insufficient accuracy associated with traditional methods. The conventional Denoising Diffusion Implicit Models (DDIM) often introduce approximation errors due to inconsistencies between the forward and reverse processes during image diffusion. BDIA enhances the precision of diffusion inversion by simultaneously considering the integration updates in both the forward and reverse directions and averaging the results [15]. Specifically, in BDIA, the update formula approximates the integral by incorporating the diffusion states at two consecutive

time steps, enabling precise diffusion inversion during time reversal. This technique not only reduces the number of function evaluations (NFE) required for both forward and reverse neural network propagation but also improves the quality of image editing and sampling.

III. Aims & Objectives

The primary aim of this study is to explore and optimize Vision Transformer (ViT) models, with the goal of identifying techniques that achieve an optimal balance between performance and resource consumption. Specifically, this research seeks to compare and analyze three different ViT model architectures to uncover the potential and limitations of reversible neural networks and BDIA technology in practical applications.

The key research questions addressed in this study are:

1) **Which of the three Vision Transformer models achieves the best balance between performance and resource consumption?**
   - This question focuses on the overall performance of the models. The goal is to compare the performance metrics (e.g., accuracy, training loss, inference time, memory usage) of the three models (ViT-small, Reversible ViT-small, BDIA ViT-small) across different datasets and tasks, to identify the model structure that performs best in resource-constrained environments.
2) **What techniques can further enhance the performance of reversible Vision Transformers?**
   - This study explores the specific roles of reversible structures and BDIA technology within ViT models, evaluating their contributions to reducing memory consumption, accelerating the training process, and improving model stability.

To achieve the research aims, this study proposes the following hypotheses and objectives:

Hypothesis 1: Reversible ViT-small can maintain or improve model accuracy while reducing memory consumption compared to the traditional ViT-small.

Objective 1: Design experiments to quantify the differences between these models in terms of memory usage, computational efficiency, and classification accuracy, and analyze the impact of the reversible structure on memory and computational resources.

Hypothesis 2: BDIA technology can further enhance the stability and accuracy of the Reversible ViT-small model, especially when training data is limited.

Objective 2: Test the BDIA ViT-small model across different datasets, evaluate the impact of BDIA technology on the model's performance during training, particularly its effect on improving validation accuracy.

Hypothesis 3: In resource-constrained hardware environments (e.g., embedded devices), reversible structures and BDIA technology can significantly reduce the resource requirements of ViT models, making the models more practical.

Objective 3: Run all models in simulated resource-constrained environments, compare their performance and resource consumption, and identify the model structure best suited for practical applications.

The study will involve training and testing the models on various datasets. The results will be analyzed to compare the models in terms of resource consumption (e.g., memory, computation time) and performance (e.g., accuracy, stability). Ultimately, the study will evaluate the practicality of reversible structures and BDIA technology, particularly in resource-constrained scenarios, and provide conclusions and suggestions for future research directions.

IV. Model Architecture & Implementation

This section provides a detailed explanation of the architecture design, mathematical methods, and technical implementation of the three models used for comparison. This includes an overview of the design logic, structure, and specific implementation steps. The details of each model will be discussed in the following subsections.

A. ViT-small Model Architecture and Implementation Details

The ViT-small model is based on the Transformer architecture, initially proposed by Dosovitskiy et al., and has demonstrated effectiveness in visual tasks. The fundamental approach of the ViT model to solving image recognition problems is to divide the input image into patches of a certain size, treat these patches as input tokens at each layer, and apply the self-attention mechanism to identify significant features in the image [2]. The ViT-small model is a concrete implementation of the ViT concept, primarily designed for training and recognition on smaller datasets. This model's architecture includes the following key components: Patch Embedding, Transformer Encoder, and MLP Head. After the input image is converted into vector data, the self-attention mechanism in the Encoder captures key features of the image, followed by the MLP classifying the image based on different features.

1) Mathematical Methods: The self-attention mechanism of the Transformer model is the most crucial part of the ViT-small model. The following section will explain the mathematical logic used. Specifically, the input image at each layer is first divided into patches of fixed size, and each patch is embedded into a high-dimensional space. This embedding vector is added to the position encoding of the patch, ultimately represented as a positional embedding, ensuring that all features of the image are captured without errors or duplication.

a) Patch Embedding: Suppose the input image has dimensions $H \times W$. The image is divided into patches of size $P \times P$, resulting in a total of $\frac{H \times W}{P^2}$ patches. Each flattened patch x_i is mapped into a high-dimensional space through a linear transformation:

$$z_i = \text{Linear}(x_i) + \text{PositionEncoding}(i)$$

It is worth noting that the PositionEncoding is implemented by the SPT class in the model and serves as a learnable positional embedding to retain the positional information of the patches.

b) Transformer Encoder: The Transformer Encoder consists of the self-attention mechanism and a feedforward neural network, with the self-attention mechanism being the core component. In each Transformer Encoder layer, the input embedding vector z is processed by the Multi-Head Self-Attention (MHSA) mechanism after Layer Normalization (LN). This facilitates global information interaction across the image, implemented by the LSA class in this model. The calculation formula for the LSA class is as follows:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q, K, V represent the Query, Key, and Value, respectively, and d_k is the dimension of the Key vector. The result is then added to the input through residual connections:

$$y = \text{LSA}(\text{LN}(x)) + x$$

The output is further processed by the feedforward neural network, implemented by the FeedForward class in this model, and again added through residual connections:

$$x = \text{FFN}(\text{LN}(y)) + y$$

2) Implementation Details: The implementation steps for the ViT-small model are as follows:

1) **Patch Embedding (SPT Module)**: The input image is first divided into patches of fixed size through the SPT module, which performs linear embedding and normalization.
2) **Positional Encoding**: The vectorized patches obtained from the previous step are added to a learnable positional embedding, forming a high-dimensional vector containing positional information.
3) **Transformer Encoder (Transformer Module)**: The high-dimensional vectors are passed through multiple Transformer Encoder layers, which include multi-head self-attention and feedforward operations to extract global features of the image.
4) **MLP Head**: The encoded feature vectors are processed by a fully connected layer to produce the final classification results.

B. Reversible ViT-small Model

The Reversible ViT-small model is fundamentally based on the ViT-small model discussed earlier. Similar to its predecessor, the basic architecture includes Patch Embedding, a Transformer Encoder, and an MLP Head. The Reversible ViT-small model first converts the input images into a series of patches through the SPT module (Shifted Patch Tokenization), then embeds these patches into a high-dimensional vector space. Building on the concept of RevNets, which are designed to reduce GPU memory usage during training by employing a reversible structure [10], this model incorporates reversible structures to enable both forward and backward propagation. According to the Reversible Vision Transformer model proposed by Karttikeya Mangalam, Haoqi Fan, et al. [14], the reversible structure divides the input tensor into two parts, I_1 and I_2, where the function F represents the multi-head self-attention mechanism and G represents the feedforward neural network (MLP block). In forward propagation:

The function F is applied to I_1 to obtain O_2:

$$O_2 = I_2 + F(I_1)$$

The function G to O_2 obtains the final output O_1:

$$O_1 = I_1 + G(O_2)$$

In the backward propagation, the inputs I_1 and I_2 are recovered from O_1 and O_2 as follows:

Recover I_1 from O_1:

$$I_1 = O_1 - G(O_2)$$

Recover I_2 from O_2:

$$I_2 = O_2 - F(I_1)$$

This approach allows the original input to be reconstructed during backward propagation without the need to store intermediate activation values. Unlike the traditional ViT-small model, the Reversible ViT-small model employs reversible blocks in the Transformer Encoder, processing these embedded vectors through the reversible Transformer encoder, and finally performing classification through the MLP Head, thereby reducing memory consumption and improving computational efficiency.

1) Implementation of the Reversible Structure: In the Reversible ViT-small model, the reversible structure within the encoder layers enables backward propagation without the need to store activations at each layer, instead reconstructing these values through computation.

Forward Propagation: In forward propagation, the input data x is processed through each reversible block to obtain the output z, just like the ViT-small model. The result of the forward propagation z is used as the input for the next layer.

Backward Propagation: The core of backward propagation is to recover the input x using the output y_2. The steps are as follows:

1) Recover the intermediate result y_1 through the backward pass of the feedforward neural network:

$$\frac{\partial L}{\partial y_1} = \frac{\partial L}{\partial y_2} + \frac{\partial(\text{FFN}(\text{LN}(y_1)))}{\partial y_1} \cdot \frac{\partial L}{\partial y_2}$$

2) Recover the input x through the backward pass of the multi-head self-attention mechanism:

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y_1} + \frac{\partial(\text{LSA}(\text{LN}(x)))}{\partial x} \cdot \frac{\partial L}{\partial y_1}$$

During backward propagation, using torch.nn's autograd automatically handles these calculations and restores the input layer by layer.

2) Differences from the Basic ViT-small Model: The primary distinction between Reversible ViT-small and the traditional ViT-small model lies in the encoder structure's reversibility. In ViT-small, the output of each layer needs to be stored, resulting in significant memory usage, especially as the model depth increases. In contrast, Reversible ViT-small reduces memory requirements significantly by implementing reversible structures, which eliminate the need to store these intermediate activations. This design provides a substantial advantage in managing the resource consumption of very deep models, thereby enhancing both the model's expressive power and training efficiency.

C. BDIA ViT-small Model

BDIA ViT-small model is built on the basic architecture of Vision Transformer (ViT) and further applies the concept of reversible neural networks along with the Bidirectional Integration Approximation (BDIA) technique. This model aims to enhance performance and memory efficiency in resource-constrained environments. The basic architecture still includes Patch Embedding, Transformer Encoder, and MLP Head. Reversible structures and the implementations of BDIA and quantization techniques have been added to the foundational structure.

1) Implementation of Reversible Structure: In the BDIA ViT-small model, the reversible structure is implemented similarly to the Reversible ViT-small model. It replaces the method of storing activations at each layer in non-reversible structures by reconstructing the activations of intermediate layers during backpropagation, thereby saving memory during training. In the BDIA ViT-small model, the reversible structure further adopts quantization techniques in both forward and backward propagation stages to achieve exact bit-level reversibility during the recomputation process. The structure and methodology will be explained in detail below.

A typical Vision Transformer block, where the input to each layer is x and the output is z, can be represented as $z = x + \text{LSA}(x) + \text{FFN}(x + \text{LSA}(x))$, which internally contains two residual connections. Following the concept from RevNets [10], the forward step at the kth time step (Equation (4)) can be viewed as an Euler integration approximation of an ODE at time step t_k [16]. This is differentiable, and from the model's perspective, the gradient flow is reversible.

2) Implementation of BDIA: The implementation of BDIA applied purely to ViT aims to optimize the model's updates using Bidirectional Integration Approximation to enhance model stability. Suppose the state update at time step t_k is represented by the following equation:

$$x_{k+1} = x_k + \Delta(t_k \to t_{k+1}|x_k) - \gamma \cdot \Delta(t_k \to t_{k-1}|x_k)$$

where $\Delta(t_k \to t_{k+1}|x_k)$ and $\Delta(t_k \to t_{k-1}|x_k)$ represent the forward and backward integration approximations, respectively. During training, the update expression can also be simply represented as:

$$x_{k+1} = \gamma x_{k-1} + (1-\gamma)x_k + (1+\gamma)h_k(x_k)$$

It is noteworthy that the choice of the γ parameter value has a certain impact. Generally, it is recommended that γ be randomly selected from $\{0.5, -0.5\}$ with equal probability for each training sample per Transformer block. It is used as a regularizer when training BDIA-ViT to ensure that neighboring Transformer blocks can change smoothly as the block index increases [17].

3) Implementation of Quantization: BDIA ViT under quantization conditions can be applied in reversible models. In a reversible model, x_k and x_{k+1} can be used to compute x_{k-1}, which means using the output of the current layer and the next layer to compute the output of the previous layer, thereby saving memory usage at the cost of more computation. The equation is:

$$x_{k-1} = \frac{x_{k+1}}{\gamma} - \frac{1-\gamma}{\gamma} \cdot x_k - \frac{1+\gamma}{\gamma} \cdot h_k(x_k)$$

However, during computation, due to the floating-point value γ, there is an increasing error in the fluctuation with time, which leads to increased error when reconstructing the previous layer's activation values, affecting the smoothness and accuracy of training. Therefore, quantization is introduced here.

The quantization operation Q_l is defined as:

$$Q_l[y] = \text{round}\left(\frac{y}{2^{-l}}\right) \cdot 2^{-l}$$

The state update equation for the quantized BDIA ViT-small model is:

$$x_{k+1} = Q_l\left[x_k + (1-\gamma) \cdot Q_l\left[\text{FFN}\left(\text{LSA}\left(\text{LN}(x)\right) + x\right)\right]\right]$$

where Q_l ensures that all intermediate activations have bit-level precision. During backpropagation, instead of storing activations, the model reconstructs the input of the previous layer through computation. The reconstruction equation is:

$$x_{k-1} = \frac{1}{\gamma}x_{k+1} - \frac{1-\gamma}{\gamma}x_k - \frac{1+\gamma}{\gamma}Q_l\left[\text{FFN}\left(\text{LSA}\left(\text{LN}(x_k)\right) + x_k\right)\right]$$

allowing online lossless reconstruction of x_{k-1} during backpropagation [17].

4) Difference from Reversible ViT-small: Compared to the Reversible ViT-small model, the BDIA ViT-small model combines BDIA techniques and quantization operations, which not only reduces memory consumption but also reduces error accumulation through precise bit-level computational accuracy during the entire inference and training stages, particularly during forward and backward propagation, resulting in higher stability.

V. Experiment & Results Analysis

The three models will be trained and verified on the CIFAR-10 and MNIST datasets respectively, which will not only obtain clearer and more reasonable performance results but also demonstrate the versatility and universality of the three models.

TABLE I
COMPARISON OF PERFORMANCE FOR THREE MODELS ON CIFAR-10

Model	Acc (ave.)	Tot Mem (MiB)	Mem (MiB/img)	GFLOPS	Param (M)	400 epochs Training Time (ave.)
ViT-small	88.26±0.31	1508	11.78	1.237	9.59	6 hours 35 minutes
Reversible ViT-small	86.99±0.24	730.38	5.71	1.237	9.59	9 hours 16 minutes
BDIA ViT-small	88.99±0.1	908.69	7.10	1.237	9.59	9 hours 51 minutes

A. Experiment

Firstly, we trained ViT-small, Reversible ViT-small, and BDIA ViT-small models on the CIFAR-10 dataset. To reduce uncontrollable errors and the impact of randomness, each model underwent three repeated experiments. The training process for each model was set to 400 epochs, ensuring that the models could fully learn and reach optimal performance. The same hyperparameters were used across all experiments: the initial learning rate was set to 1e-4, the optimizer used was SETAdam [15], and the batch size was set to 128. The image patches were uniformly resized to 32, and each Transformer block was configured with the same number of attention heads and depth. Additionally, identical image preprocessing was applied to create the same training set and test set, which were then used by all three models. This ensures that the models were trained and tested under exactly the same conditions, eliminating other random factors. As a result, the comparative results obtained are reliable and provide a convincing assessment of the performance differences between the three models under identical conditions.

Subsequently, we also trained ViT-small, Reversible ViT-small, and BDIA ViT-small models on the MNIST dataset to further verify whether the results are consistent with those obtained on CIFAR-10. Similarly, to minimize uncontrollable errors caused by randomness, each model underwent three repeated experiments. The training process for each model was set to 50 epochs, which is sufficient for achieving optimal performance on the MNIST dataset. The same hyperparameters were applied across all experiments same to the experiment on CIFAR-10, only the image patches were uniformly resized to 28.

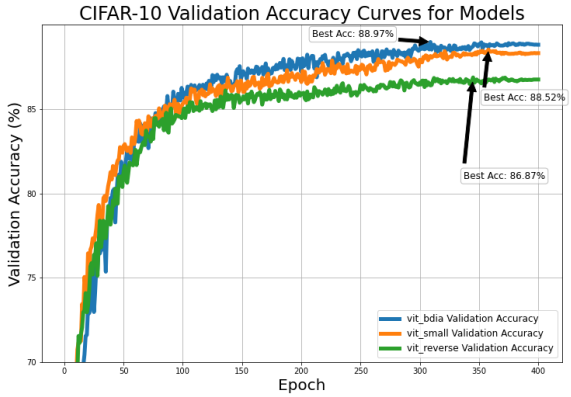

Fig. 1. Three models' Validation Accuracy comparison on CIFAR-10 (The most representative result of the three runs)

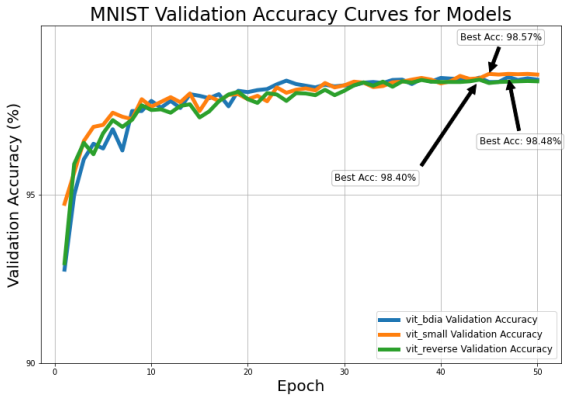

Fig. 2. Three models' Validation Accuracy comparison on MNIST (The most representative result of the three runs)

Figure 1 1presents a comparative analysis of validation accuracy obtained from one of the most representative training runs among the three conducted on the CIFAR-10 dataset. In this particular run, the BDIA ViT-small model demonstrated the most outstanding performance, achieving a final validation accuracy of 88.97%. The ViT-small model followed closely, with a validation accuracy of 88.5%. The Reversible ViT-small model, on the other hand, showed slightly lower performance, with a peak validation accuracy of 86.87%. Similarly, Table 1 I reinforces this conclusion, showing that the BDIA ViT-small model achieves the highest average validation accuracy, while the Reversible ViT-small model lags slightly behind. From the overall trend, it is evident that the BDIA ViT-small model initially lags slightly in the early stages of training but quickly surpasses the other two models as the training progresses, indicating its competitive advantage in improving image recognition accuracy with deeper learning. Additionally, the BDIA ViT-small model maintains relatively stable accuracy once it reaches higher levels, indicating not only superior accuracy during the learning process but also better stability in maintaining that accuracy. In contrast, the Reversible ViT-small model exhibits a slower learning curve with limited improvement in the later stages of training. The ViT-small model, which performs moderately, shows a rapid increase in accuracy during the early stages of training, but its final accuracy is slightly lower compared to the BDIA ViT-small model.

On the MNIST dataset, as shown in Figure 2 2, the validation accuracy from one of the most representative training runs further confirms the performance comparison among the three models. The BDIA ViT-small model achieved the highest

validation accuracy of 98.57%, followed by the ViT-small model with 98.48% validation accuracy, and the Reversible ViT-small model again trailing slightly with a peak validation accuracy of 98.40%. The similar trends in curve shapes and the distribution of the best validation accuracies across the three models on both CIFAR-10 and MNIST datasets further substantiate the correctness of the performance comparison among the models.

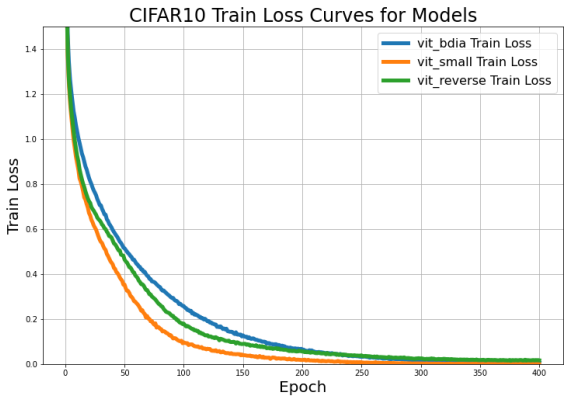

Fig. 3. Three models' Training loss comparison on CIFAR-10

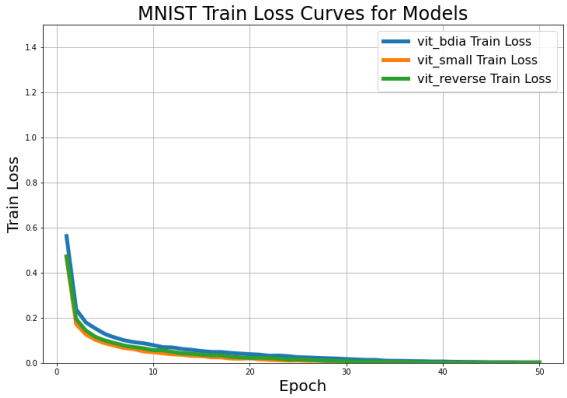

Fig. 4. Three models' Training loss comparison on MNIST

Figure 3 3 presents the training loss curves for the three models on the CIFAR-10 dataset. Throughout the training process, the ViT-small model shows the fastest decline in loss, indicating a higher convergence rate. This suggests that the ViT-small model, under the current hyperparameter settings, can quickly learn the features of the dataset and optimize its loss function to reduce errors. The Reversible ViT-small model shows a slightly slower loss reduction compared to ViT-small, with a higher loss in the later stages of training. Additionally, it experiences a noticeable fluctuation during training, indicating potential instability in the training process and a higher risk of overfitting. The BDIA ViT-small model has a slower convergence rate initially, but its loss gradually approaches that of the ViT-small model in the later stages. Notably, the BDIA ViT-small model exhibits a more stable learning curve during training, suggesting that the quantization operation and BDIA technique positively contribute to the stability of model training. Although the final training loss is slightly higher, its performance in validation accuracy compensates for this.

In Figure 44, which shows the training on the MNIST dataset, the training loss curves for the three models follow a similar trend. The Reversible ViT-small and ViT-small models have almost identical loss curves in the initial stages, indicating similar learning capabilities at the beginning of training. In contrast, the BDIA ViT-small model starts with a higher initial loss. However, in the later stages of training, the BDIA ViT-small model's loss slightly surpasses the other two models, demonstrating better training results. Furthermore, the training loss curves for all three models on the MNIST dataset are relatively smooth and converge quickly, indicating that on this simpler dataset, all three models can effectively learn and quickly optimize their loss functions.

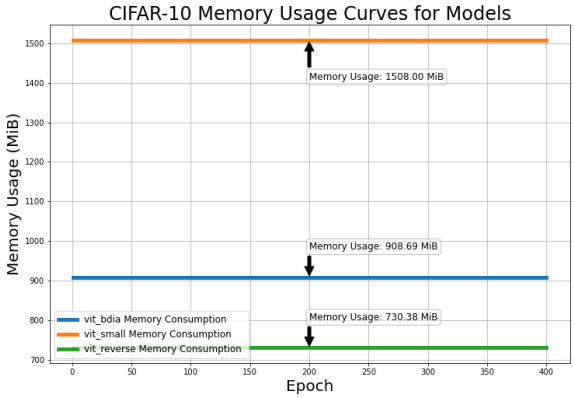

Fig. 5. Three models' Memory measurement comparison on dataset CIFAR-10

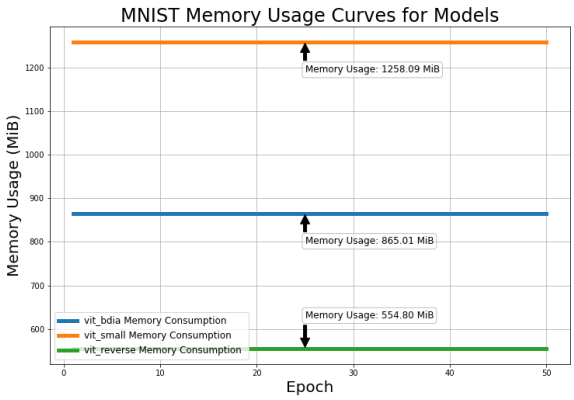

Fig. 6. Three models' Memory measurement comparison on dataset MNIST

Figure 5 5shows the memory usage of the three models

during training on the CIFAR-10 dataset. The ViT-small model has the highest memory consumption, reaching 1508.00 MiB. This is primarily due to the ViT-small model needing to store a large number of intermediate activations during computation, leading to higher memory usage. In contrast, the Reversible ViT-small model, which utilizes a reversible network structure that does not require saving all intermediate activations, has the lowest memory consumption, only 730.38 MiB, significantly reducing memory usage. The BDIA ViT-small model consumes 908.69 MiB of memory, falling between the ViT-small and Reversible ViT-small models. Although the BDIA ViT-small model employs quantization operations and BDIA techniques, it strikes a balance between accuracy and memory consumption, resulting in relatively moderate memory usage.

Figure 6 6presents the memory usage of the three models during training on the MNIST dataset, which shows a similar trend to that observed on the CIFAR-10 dataset. Due to the smaller size of the dataset, the memory consumption of all three models is reduced on the MNIST dataset. The ViT-small model still exhibits the highest memory consumption, reaching 1258.09 MiB, consistent with its performance on CIFAR-10. The Reversible ViT-small model has the lowest memory consumption on the MNIST dataset, at 554.80 MiB, continuing to demonstrate its memory efficiency advantage. The BDIA ViT-small model consumes 865.01 MiB on the MNIST dataset, showing good memory efficiency, slightly higher than the Reversible ViT-small but still significantly more efficient compared to the ViT-small model.

Additionally, as observed in Table 1I, both ViT models with reversible structures exhibit significant reductions in memory consumption. The experimental tests were conducted on a platform using an NVIDIA RTX 4060 laptop GPU with 8GB of VRAM, and the batch size for the experiments was uniformly set to 128. When considering the maximum batch size under the 8GB limitation, the ViT-small model can only set the batch size to around 680, while the Reversible ViT model can be set to 1400, and the BDIA ViT model can be set to around 1100, both achieving significant improvements. However, due to the simpler model structure, ViT-small demonstrates a significant increase in training efficiency. Under the current training conditions, it can reduce the training time by approximately 3 hours compared to the two ViT models with reversible structures.

B. Analysis and Insights

Reasons for the Lower Accuracy of the Reversible ViT Model:

The core feature of the Reversible ViT model is its use of a reversible structure. The main advantage of this structure is the reduction of memory consumption. However, this may directly lead to a decrease in model accuracy. The reversible structure requires recalculating the intermediate activation values during backpropagation instead of directly reading them from memory. Since the code utilizes float variables for storing activation values during the calculation process, particularly, as the model depth increases, the cumulative effect of errors in gradient computation during backpropagation may cause the model to struggle to capture subtle feature differences, leading to more significant fluctuations in accuracy when reaching higher precision during training and ultimately affecting the model's performance.It could become more significant in complex tasks or large-scale datasets. Therefore, while reversibility aids in memory optimization, this optimization may come at the cost of accuracy in certain situations. This suggests that when designing efficient models, careful consideration is needed for the trade-off between memory and accuracy. This is also one of the reasons for attempting to introduce BDIA (Bidirectional Integration Approximation) into the reversible structure.

Advantages and Better Performance of BDIA ViT Model with Quantization:

The BDIA ViT model introduces BDIA (Bidirectional Integration Approximation), a technique that combines forward and backward integration, enabling the model to avoid error accumulation from single-direction integration during parameter updates. In the ViT model, this helps mitigate or even completely eliminate the impact of errors during the forward and backward propagation processes. At the same time, the introduced quantization operation further stabilizes and secures the model's computation process. Quantization can have a regularization effect, preventing the model from overfitting during training, thus enhancing the model's generalization ability. Consequently, although the BDIA ViT may exhibit slightly higher training loss in the early and even middle stages of training, the accuracy advantage of its computation will gradually emerge with more iterations, resulting in higher validation accuracy and broader applicability in practical scenarios.

Why BDIA ViT Consumes More Memory Than Reversible ViT:

This is inevitable. Although the BDIA ViT model introduces a reversible structure to reduce memory consumption, the additional BDIA computations and quantization operations necessarily increase memory usage. BDIA's bidirectional integration requires additional calculations during both forward and backward propagation, so it requires storing more side information to accurately reconstruct the activation values during backpropagation. These extra computation and storage costs result in higher memory usage for BDIA ViT compared to the Reversible ViT model. This overhead is justifiable. Giving the consideration of improvements in accuracy and stability, this is a worthwhile trade-off. This comparison highlights that in designing memory-sensitive deep learning models, a moderate increase in memory overhead can yield significant performance gains.

Local vs. Global Optima in Deep Models:

The rapid convergence of the ViT-small model in the early stages of training suggests that it is prone to finding local optima. This is often due to the simplicity of the model's structure and the smaller parameter space, making it easier to find a local optimal solution under specific hyperparameter settings. However, the training curves of Reversible ViT-small and BDIA ViT-small indicate that while their initial conver-

gence is slower, they exhibit better stability and accuracy in the later stages, demonstrating their ability to gradually approach a globally optimal solution in a more complex parameter space. This phenomenon suggests that although simpler models may perform well in the short term, incorporating more computational techniques and structural innovations (such as BDIA) might be more advantageous in finding better solutions during long-term training, especially for more complex tasks.

C. Integration of Momentum

In solving the problem of deep models getting stuck in local optima, momentum has always been widely mentioned and adopted as a method. Momentum not only plays a role in optimizers but is also a valuable attempt to help models achieve smooth convergence and escape local optima in reversible models' propagation. As demonstrated in the research by Prin Phunyaphibarn and Junghyun Lee, the impact of momentum in gradient descent is significant, particularly in triggering the "catapult" effect during training. At higher learning rates, momentum gradient descent exhibits a more significant catapult effect, which positively influences the training trajectory by helping the model navigate through the minima that can be achieved with the current gradient propagation. This enhances the "sharpness" of the model, thereby improving overall stability [18].

We conducted some experiments to integrate momentum into the ViT model, attempting to introduce momentum during both forward and backward propagation in the Reversible ViT-small model. The approach involved applying the LSA function to the output result x of the previous layer, adding it to the residual, and then applying the FFN function along with the sum of the residuals. The process can be mathematically described as follows:

$$y = \text{LSA}(\text{LN}(x)) + x$$

$$z = \text{FFN}(\text{LN}(y)) + y$$

Furthermore, building on the ideas from Sander's team [13], we introduced a separate momentum component m that is added to the sum of the residuals during forward propagation:

$$z = \text{FFN}(\text{LN}(y)) + y + m$$

During backpropagation, we ensure the correct gradient propagation by recovering the momentum component through differential calculations. The benefit of this approach is that it leverages momentum characteristics in deep models to smooth the training process further and improve the impact on accuracy.

However, the introduction of momentum in the model's implementation brings two issues that need to be solved. First, momentum accumulates historical information and requires constant updating during iterations. However, the method of calculating momentum during iterations can lead to unpredictable errors in gradient propagation, such as gradient explosion or vanishing. Updating momentum requires more complex methods and logic to ensure its value remains within an acceptable range, but such handling may undermine the original purpose of using momentum. Second, the basic Reversible ViT model can run correctly without errors during the validation phase. However, due to the introduction of momentum, model validation requires more consideration and operations. The current issue is that while the model usually performs during the training phase, it encounters problems during validation. This suggests that the features learned by the model are incorrect, and momentum may have compromised the model's correctness to some extent. This is likely due to errors in recomputing gradients and momentum during backpropagation. A proposed solution is to store the momentum at each step during training for use in the reversible structure, but this contradicts the original intention of saving memory in the reversible structure. Therefore, the idea of introducing momentum into the Reversible ViT still requires further consideration and research to address these challenges.

VI. Conclusion

This study focuses on the performance optimization and resource consumption control of Vision Transformer (ViT) models. We can draw several clear conclusions by comparing the performance of three models—ViT-small, Reversible ViT-small, and BDIA ViT-small on the CIFAR-10 and MNIST datasets. First, although the Reversible ViT-small model has significant advantages regarding memory consumption, its accuracy is slightly lower than that of the standard ViT-small model due to precision loss in gradient backpropagation caused by the reversible structure. The BDIA ViT-small model, by incorporating Bidirectional Integration Approximation (BDIA) and quantization operations, not only surpasses Reversible ViT-small in accuracy and stability but strikes a balance between performance and resource usage in terms of memory consumption.

The experimental results demonstrate that the BDIA ViT model improves accuracy through technical implementation without significantly increasing memory overhead, making it more applicable in resource-constrained environments. However, there is still room for further optimization regarding memory usage. For instance, further processing of the quantized parameters after quantization to avoid complex storage is a potential direction for future research and implementation. Although the introduction of momentum into the Reversible ViT faced some challenges, it remains a promising approach, providing new directions for further optimizing reversible models in the future.

This study offers robust theoretical support and practical guidance for achieving efficient resource management and model optimization in complex tasks through an in-depth analysis of different ViT model architectures. In practical applications, these technologies can be further explored on larger-scale datasets and in more diverse scenarios to promote the widespread adoption of Vision Transformer models.

VII. Declarations

Declaration of Originality. I am aware of and understand the University of Exeter's policy on plagiarism and I certify that this assignment is my own work, except where indicated by referencing, and that I have followed the good academic practices.

Declaration of Ethical Concerns. This work does not raise any ethical issues. No human or animal subjects are involved neither has personal data of human subjects been processed. Also no security or safety critical activities have been carried out.

References

[1] M. Zhu, Y. Tang, and K. Han, "Vision transformer pruning," 2021. [Online]. Available: https://arxiv.org/abs/2104.08500

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[3] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers amp; distillation through attention," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 10 347–10 357. [Online]. Available: https://proceedings.mlr.press/v139/touvron21a.html

[4] H. Touvron, M. Cord, and H. Jégou, "Deit iii: Revenge of the vit," in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 516–533.

[5] K. Islam, "Recent advances in vision transformer: A survey and outlook of recent work," 2023. [Online]. Available: https://arxiv.org/abs/2203.01536

[6] A. Khan, Z. Rauf, A. Sohail, A. R. Khan, H. Asif, A. Asif, and U. Farooq, "A survey of the vision transformers and their cnn-transformer based variants," *Artificial Intelligence Review*, vol. 56, no. 3, pp. 2917–2970, 2023. [Online]. Available: https://doi.org/10.1007/s10462-023-10595-0

[7] N. Park and S. Kim, "How do vision transformers work?" 2022. [Online]. Available: https://arxiv.org/abs/2202.06709

[8] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 10 012–10 022.

[9] H. Touvron, M. Cord, A. El-Nouby, J. Verbeek, and H. Jégou, "Three things everyone should know about vision transformers," 2022. [Online]. Available: https://arxiv.org/abs/2203.09795

[10] A. N. Gomez, M. Ren, R. Urtasun, and R. B. Grosse, "The reversible residual network: Backpropagation without storing activations," *Advances in neural information processing systems*, vol. 30, 2017.

[11] B. Chang, L. Meng, E. Haber, L. Ruthotto, D. Begert, and E. Holtham, "Reversible architectures for arbitrarily deep residual neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[12] A. Gholami, K. Keutzer, and G. Biros, "Anode: Unconditionally accurate memory-efficient gradients for neural odes," *arXiv preprint arXiv:1902.10298*, 2019.

[13] M. E. Sander, P. Ablin, M. Blondel, and G. Peyré, "Momentum residual neural networks," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9276–9287.

[14] K. Mangalam, H. Fan, Y. Li, C.-Y. Wu, B. Xiong, C. Feichtenhofer, and J. Malik, "Reversible vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 830–10 840.

[15] G. Zhang, J. P. Lewis, and W. B. Kleijn, "Exact diffusion inversion via bi-directional integration approximation," 2023. [Online]. Available: https://arxiv.org/abs/2307.10829

[16] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf

[17] G. Zhang, J. P. Lewis, and W. B. Kleijn, "On exact bit-level reversible transformers without changing architectures," 2024. [Online]. Available: https://arxiv.org/abs/2407.09093

[18] P. Phunyaphibarn, J. Lee, B. Wang, H. Zhang, and C. Yun, "Gradient descent with polyak's momentum finds flatter minima via large catapults," 2024. [Online]. Available: https://arxiv.org/abs/2311.15051