Frozen Language Models are Gradient Coherence Rectifiers in Vision Transformers

Lichen Bai[1*], Zixuan Xiong[1,3*], Hai Lin[1,2], Guangwei Xu[3], Xiangjin Xie[3], Ruijie Guo[3], Zhanhui Kang[4], Hai-Tao Zheng[1,2†], Hong-Gee Kim[5]

[1]Shenzhen International Graduate School, Tsinghua University
[2]Pengcheng Laboratory
[3] Alibaba Cloud Computing
[4] Machine Learning Platform Department, Tencent
[5] Seoul National University
[1]{blc22, xzx22}@mails.tsinghua.edu.cn, [3]{kunka.xgw, xiexiangjin.xxj, ruijie.guo}@alibaba-inc.com
†zheng.haitao@sz.tsinghua.edu.cn

Abstract

Large language models (LLMs) have demonstrated remarkable performance in multimodal tasks even with frozen LLM Block and only a few trainable parameters. However, the underlying mechanisms of how LLMs enhance multimodal performance remains unclear. In this work, we focus on the phenomenon that "Merely concatenating a frozen LLM block to the Vision Transformer (ViT) encoder can yield significant performance enhancements. Moreover, the choice of LLM block and insertion position can have a substantial impact, leading to varying degrees of improvement". We analyze the optimization of the training process from the perspective of gradient dynamics and find that frozen LLM blocks act as gradient coherence rectifiers, aligning the gradients of different samples more closely during training. Furthermore, we demonstrate that the representation similarity between the inserted LLM block and the adjacent ViT block influences performance, with greater similarity tending to yield larger positive gains. Through these findings, we can justify the selection of suitable LLM blocks to be inserted at appropriate positions, and introduce additional gradient backpropagation paths by incorporating LLM blocks, could improve the performance of vanilla ViT through the rectification effect of gradient consistency during the training process, without the need to add LLM blocks during inference. Our experiments demonstrate the effectiveness of this strategy, making the practical application of the gradient rectification effect feasible.

Introduction

Recently, with the continuous increase in computational power and data volume, modern large language models (LLMs) have demonstrated a trend towards unified processing in natural language processing (NLP) tasks (Touvron et al. 2023; Workshop et al. 2022). Furthermore, even when trained on pure text, LLMs possess strong multimodal capabilities. Through minimal-cost fine-tuning, they can serve as the core processor for image (Li et al. 2023), video (Zhang, Li, and Bing 2023), point cloud data (Yu et al. 2022) and beyond. To some extent, they have been shown to possess general representations (Huh et al. 2024).

Specifically, Pang et al. (2023) experimentally demonstrated that stacking frozen LLM blocks on top of a visual encoder can significantly enhance the performance of purely vision tasks, such as image classification (Touvron et al. 2021) and video understanding (Tong et al. 2022). They attribute this phenomenon to the "information filtering hypothesis," which posits that pre-trained LLM transformer blocks enhance the impact of informative visual tokens. However, this does not empirically reveal the role of LLM Blocks during training.

In this work, from the perspective of gradient dynamics, we analyze the optimization process of gradient descent and find that frozen LLM blocks act as gradient rectifiers, aligning gradients of different samples more closely during training, see Fig. 1. Chatterjee and Zielinski (2022) argue that the greater gradient similarity between different samples makes the optimization process focus on learning common patterns while minimizing the impact of those outliers. In order to better observe the gradient changes during training, we introduce the Gradient Signal-to-Noise Ratio (GSNR) to represent gradient coherency (Sun et al. 2023), measuring the generalization performance of the model.

Our experiments verify that the frozen LLM block has a certain gradient coherence rectification effect. After it is inserted into the ViT model, the average GSNR of the ViT model increased significantly. Furthermore, our layer-wise analysis reveals that the gradients of those layers closer to the LLM blocks exhibits higher coherence. This suggests that the gradient rectification effect of LLM blocks primarily influences the nearby parameters.

Moreover, we observe that inserting different LLaMA Blocks at various positions within the ViT significantly influences the performance gains. Inspired by Huh et al. (2024), we employ Mutual-KNN to analyze the representation similarity between different layers of ViT and LLaMA. Our experiments indicate that when the inserted LLaMA

*These authors contributed equally.
†Corresponding author.

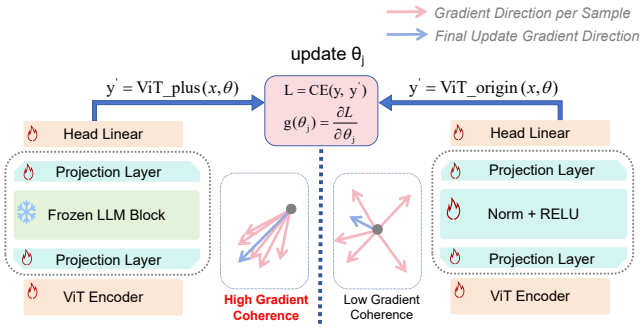

Figure 1: Illustration of gradient rectification effect. For each parameter θ_j, when the loss function L is backpropagated through the frozen LLM block, it enhances the coherence of gradients of different samples. As a result, the network learns more general patterns within the dataset, thereby enhancing the ability of generalization.

Block exhibits higher representation similarity with the adjacent ViT Block, the gradient rectification effect yields greater gains. Conversely, when there are larger representation differences, the inserted LLaMA Block often leads to a reduction in gains and may even result in decreased performance.

Finally, directly utilizing this gradient rectification effect by inserting LLM Blocks significantly increases inference costs (e.g., 8.5G FLOPs to 90.11G FLOPs). And based on our findings, we propose an auxiliary training method that can derive positive gains from the LLM Block without incurring additional inference costs. During training, we integrate the LLaMA Block into the gradient backpropagation path, enforcing stronger gradient consistency. During inference, we retain only the original forward inference path, discarding most of the parameters used during training (e.g., LLM Block, Adapter Linear, etc.). Our experiments demonstrate the effectiveness of this strategy, making the practical application of the gradient rectification effect feasible.

In summary, the contributions are as follows:

- Our findings reveal that inserting frozen LLM blocks into ViT can significantly rectify the gradient coherence. And gradients of those layers closer to the LLM blocks exhibits higher coherence.
- We propose a strategy for utilizing the gradient rectification effect. By using Mutual-KNN heatmaps, the suitable combination of LLaMA Block and insertion position in the ViT can be selected. The more similar their representations, the greater the performance gains tend to be achieved.
- We propose an auxiliary training method that harnesses the effect of gradient coherence rectification. This approach enables us to achieve significant generalization improvements without the need to add extra LLM blocks or alter the model structure during inference.

Related Work

Large Language Models Beyond Linguistics

In recent years, with the expansion of training data and the increase in model parameters, large language models like GPT4 (Achiam et al. 2023), LLaMA (Touvron et al. 2023), OPT (Zhang et al. 2022), BLOOM (Workshop et al. 2022), PaLM (Chowdhery et al. 2023) achieve considerable success in linguistics. Furthermore, LLMs demonstrate the potential as versatile components across various tasks (Merullo et al. 2022). Consequently, Blip2 (Li et al. 2023) uses projection, while Flamingo (Alayrac et al. 2022) employs cross-attention, to incorporate visual information into LLMs. Expanding on this, a variety of modalities including speech (Zhang et al. 2023), videos (Zhang, Li, and Bing 2023), 3D point clouds (Hong et al. 2023), and graphs (Wu et al. 2023) can be converted into tokens comprehensible by LLMs. Additionally, Pang et al. (2023) demonstrate that frozen, pre-trained LLM blocks could function as a general-purpose encoder, enhancing the expressive capacity of ViTs (Dosovitskiy et al. 2020).

Gradient Dynamics

Generalization is considered the key to superior performance in neural networks (Zhang et al. 2021a). Some studies suggest that stronger coherence among gradients enhances the generalization capability of a model. For instance, the concept of gradient signal-to-noise ratio (GSNR) (Liu et al. 2019) characterizes gradient coherence from both theoretical and empirical perspectives. Subsequently, smoothing gradients to reduce gradient variance of samples emerges as a novel optimization approach, applied in tasks such as neural architecture search (Sun et al. 2023), domain generalization (Michalkiewicz et al. 2023) and federated learning (Zhang et al. 2021b).Fort et al. (2019) introduce stiffness to measure the ability of generalization which focuses on how gradients in one sample affect loss changes in another. Higher stiffness indicates better performance. Additionally, Chatterjee and Zielinski (2022) introduce the normalized average dot product of gradients as a measure of coherence. Furthermore, they offer explanations grounded on gradient coherence for techniques like early stopping, L1/L2 norms, and dropout.

Auxiliary Training

Enhancing optimization by merging gradients from the main task and auxiliary tasks allows for overcoming local optima point (Ruder 2017). For example, Foret et al. (2020) introduce the SAM loss to promote joint optimization of auxiliary tasks and achieve superior performance. Tiwari and Shenoy (2023) introduce feature sieving through auxiliary training in shallow layers to encourage the forgetting of shallow features. Effective auxiliary training acts as implicit regularization for features, cultivating stronger representations in the network to improve generalization (Liebel and Körner 2018; Bardes, Ponce, and LeCun 2022; Taha et al. 2021). Inspired by this, we incorporate the LLM blocks into auxiliary training, leveraging the gradient rectification effect to enhance

optimization coherence between samples and improve the ability of generalization.

Representation Convergence

There is a growing representation similarity between neural networks as the model capacity and data scale increase. Merullo et al. (2022) prove that ultra-large linguistic and vision models exhibit isomorphic properties between text and vision features, even when unaligned. Kornblith et al. (2019) introduced Centered Kernel Alignment (CKA) to quantify this representation similarity, suggesting that similar knowledge can be learned across different architectures, data modalities, and objectives functions. Building on this, Park et al. (2023) proposed a Mutual-KNN clustering-based similarity method, which relaxes CKA's strict alignment requirements and yields more reliable measurements. Inspired by Huh et al. (2024), who argue that models across different modalities reflect world knowledge in different ways, we validate that general representations in LLM blocks enhance downstream task performance.

Preliminaries

In this section, we introduce prerequisites and notations.

Layer-Wise GSNR

We first introduce GSNR and propose a layer-wise gradient analysis method. Assuming a batch size of B and data distribution follows $\mathbb{Z}$, with each training data pair represented as $(\mathbf{x}, y)$, training the model $\hat{y}$ = model($\mathbf{x}$, $\boldsymbol{\theta}$) under the objective function L. For the i-th sample, it follows as

$$g_i(\boldsymbol{\theta}) = g(\mathbf{x}_i, y_i, \boldsymbol{\theta}) = \frac{\partial L(\hat{y}_i, y_i)}{\partial \mathbf{x}_i}. \tag{1}$$

Specifically, for the j-th parameter $\boldsymbol{\theta}_j$, we can obtain its square of the expected value $\mathbb{E}^2(g(\boldsymbol{\theta}_j))$ and variance $\text{Var}(g(\boldsymbol{\theta}_j))$ within a mini batch, enabling the calculation of the individual parameter's $gsnr(\boldsymbol{\theta}_j)$ as

$$gsnr(\boldsymbol{\theta}_j) = \frac{\mathbb{E}^2(g(\boldsymbol{\theta}_j))}{\text{Var}(g(\boldsymbol{\theta}_j))} = \frac{\mathbb{E}^2(g(\boldsymbol{\theta}_j))}{\mathbb{E}(g^2(\boldsymbol{\theta}_j)) - \mathbb{E}^2(g(\boldsymbol{\theta}_j))}. \tag{2}$$

Due to differing GSNR behaviors across layers (e.g., scale discrepancies), full-parameter statistics might overlook crucial details. Hence, departing from the traditional sample-wise approach, we opt for a layer-wise analysis to track the GSNR's temporal dynamics within each ViT block.

To obtain accurate and smooth results, we aggregate the performance of each epoch over T batches and compute the average GSNR value. Specifically, for the l-th ViT block at the n-th epoch, its layer-wise gsnr values are computed as

$$gsnr_l^n = \frac{1}{T}\sum_{t=1}^{T} \frac{\sum_{\boldsymbol{\theta}_j \in \text{Block}_l} gsnr(\boldsymbol{\theta}_j)}{\text{Params}(\text{Block}_l)}. \tag{3}$$

A higher $gsnr_l^n$ indicates that at the n-th epoch, the gradients at l-th block of the ViT model are more consistent, reflecting enhanced generalization.

Mutual K-Nearest Neighbor Metric

Following Huh et al. (2024), to compute the representation similarity between two models f and g, for a mini batch $\{x_i, y_i\}_{i=1}^b$ sampled from the multi-modality distribution $\mathbb{D}$ (e.g., image-caption dataset), where x_i represents text data and y_i represents the corresponding vision data, we obtain the latent representations $\Phi_i = f(x_i)$ and $\psi_i = g(y_i)$. Thus, we have text feature collection $\Phi = \{\phi_1, \cdots, \phi_b\}$ and vision feature collection $\Psi = \{\psi_1, \cdots, \psi_b\}$.

For each feature pair $\{\phi_i, \psi_i\}$, we compute the respective k nearest neighbor sets $S(\phi_i, k)$ and $S(\psi_i, k)$. Then we measure its average intersection as

$$\text{m}_{NN}(\phi_i, \psi_i) = \frac{1}{k}|S(\phi_i, k) \cap S(\psi_i)|, \tag{4}$$

where $|\cdot|$ is the size of the intersection. As m_{NN} increases, it indicates greater similarity between the representations of f and g.

LLM As Gradient Rectifiers

In this section, we explore the role that the LLM Block plays during training from a dynamic perspective.

Following prior works (Pang et al. 2023; Touvron et al. 2021), we utilize ViT-S (Dosovitskiy et al. 2020) as the visual encoder backbone for image classification task, and Block_l is the l-th vision transformer block and the whole backbone has L layers. Then, final predictions of vanilla backbone are obtained using the classification head layer, as shown in Eq. (5):

$$\begin{aligned} z^{l+1} &= \text{Block}_l(z^l), \\ y &= \text{Head}(z^L). \end{aligned} \tag{5}$$

For combining LLM block, we select the last block of LLaMA-7B, freeze its parameters and insert it at the end of the vvisual encoder just like Fig. 1, denoted as LLM. We then use a linear layer to align the dimensions, as shown in Eq. (6):

$$\begin{aligned} \tilde{z}^L &= \text{MLP}(\text{LLM}(\text{MLP}(z^L))), \\ \tilde{y} &= \text{Head}(\tilde{z}^L). \end{aligned} \tag{6}$$

To analyze gradient changes, we choose the Deit architecture (Touvron et al. 2021) and train it on CIFAR-100 (Krizhevsky et al. 2009) for 300 epochs. For specific details and more tasks, see supplemental material.

LLM Block Enhances Gradient Coherence

The insertion of LLM Block enhances gradient consistency throughout the training process, as shown in Fig. 2.

This implies that integrating the LLM block effectively captures common patterns in the training data, thereby obtaining more powerful features for general parameter optimization. We call this phenomenon the "gradient rectification effect" as it vividly illustrates the role of LLMs in vision downstream tasks by making gradients within a batch more consistent, as illustrated in Fig. 1.

Additionally, in the later stages of training, as the model converges and the benefits of gradient coherence updates become limited, the GSNR stabilizes at a specific value.

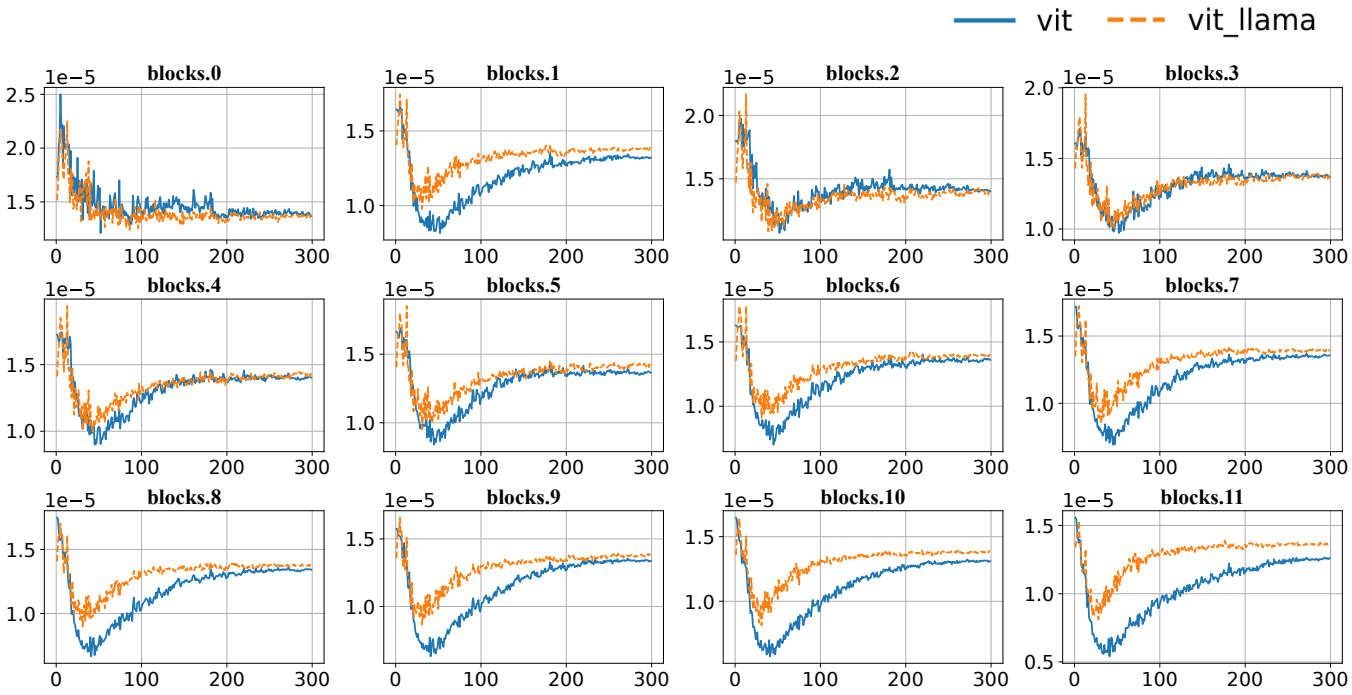

Figure 2: The trend of GSNR with training process. Horizontal axis: Number of training epochs; Vertical axis: GSNR Value. After inserting the frozen LLM Block at the end of the Vision Encoder, the GSNR is higher than in the vanilla ViT, indicating stronger gradient consistency. This effect is consistently observed across various tasks and datasets, more scenarios in supplemental material.

More Closer, More Gains

Next, to further investigate the relationship between LLM and gradient consistency, we insert LLM Block at the beginning, middle layer, and end of the ViT.

We define the average layer-wise GSNR throughout the whole training process of N epoch as

$$gsnr_l = \frac{1}{N}\sum_{n=1}^{N} gsnr_l^n, \tag{7}$$

where $gsnr_l$ represents the average gradient consistency of the l-th layer in ViT. To clearly demonstrate the impact of the LLM, we calculate the GSNR difference between the ViT with the inserted LLM Block and the corresponding layers of the vanilla ViT. A larger difference indicates a stronger consistency gain provided by the LLM for that layer.

In Fig. 3, we illustrate the impact of inserting LLM Blocks at various positions within the ViT on the consistency gain for each layer.

Due to gradient backpropagation, the inserted LLM Block mainly affects the parameters of preceding ViT layers, which also explains why inserting the LLM at the beginning results in a negative gain, as all ViT Blocks are positioned after the LLM, preventing the gradient from being rectified through the LLM. Additionally, we found that parameters closer to the llm rectifier are more likely to benefit from coherent gradients.

Model-Size	LLM@560M	LLM@1.1B	LLM@3B	LLM@7B
vit@5.7M	0.0864	0.0926	0.1185	0.1231
vit@22M	0.0924	0.0982	0.1268	0.1315
vit@86M	0.0955	0.1004	0.1286	0.1330

Table 1: Larger language models and vision transformers exhibits higher representation similarity.

Similar Representations Yield Gains

In this section, we aim to determine what properties of the LLM Block enable it to function as a general gradient rectifier. And we elucidate this from the perspective of representation similarity.

Similarity Between Visual and Linguistic Features

Inspired by Huh et al. (2024) who argue that larger models have more similar representations, our experiments also confirm that increasing the scales of vision and linguistic models leads to more similar representations, see Tab. 1.

And Pang et al. (2023) indicates that larger pre-trained LLM Blocks provide greater benefits, whereas very small block can result in negative gains, see Tab. 2.

On one hand, larger models show higher representation similarity. On the other hand, larger models also achieve greater performance gains from LLMs. Hence we argue that the greater similarity between LLM blocks and and the ViT

[1]The result is quoted from Pang et al. (2023).

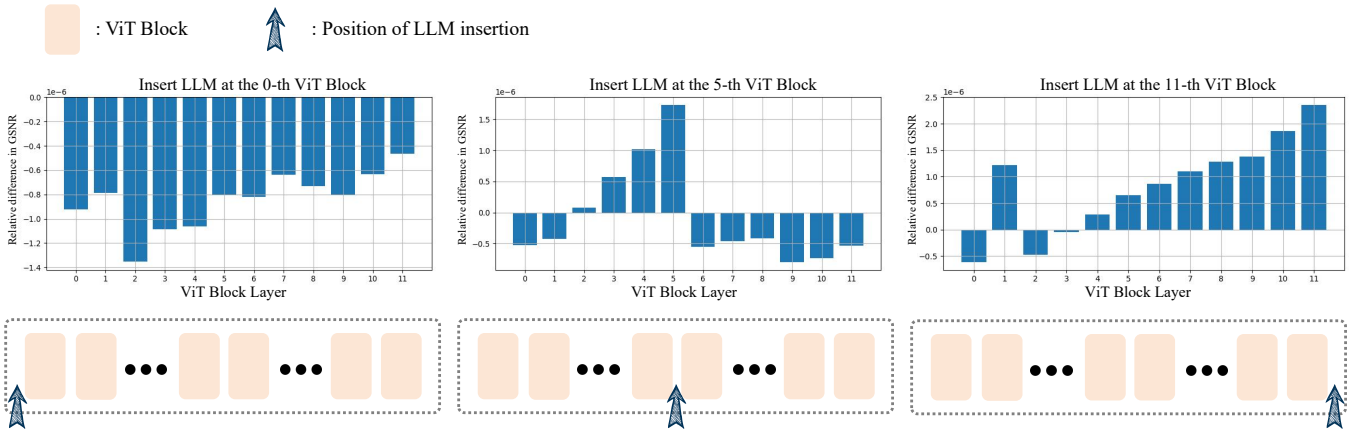

Figure 3: The difference in GSNR compared to the vanilla version when LLM blocks are inserted at the beginning, middle, and end positions. **(Right)** When inserted at the end position, the LLM block enhances GSNR for all preceding ViT blocks, with closer parameters benefiting more. **(Middle)** When inserted at the middle position (e.g, 5-th ViT layer), the coherence gradient backpropagation primarily benefits the parameter updation in the first half layers. **(Left)** When inserted at the 0-th layer, the LLM block offers minimal gains as it doesn't impact any ViT blocks. For more details, please refer to the supplemental material.

Acc@1	+Opt-350M	+Opt-1.3B	+Opt-2.7B	+Opt-6.7B
ViT-S	71.56(-3.69)	75.62(+0.37)	75.74(+0.54)	76.29(+1.04)

Table 2: With larger transformer block inserted, ViT-S exhibits better performance in ImageNet.[1]

Block near the insertion position can lead to larger performance gains.

More Similarity, More Gains

To validate this, we calculate the Mutual-KNN score between each block of the pre-trained LLaMA-7B and pre-trained ViT-S layer by layer to quantify their similarity. It can be observed in Fig. 4 that the representation similarity between later layers is higher, indicating that deeper layers tend to learn more general features which is also consistent with the conclusions drawn by Jin et al. (2024).

As shown in Tab. 3, we perform multiple experiments on CIFAR-100, inserting LLM blocks from layers {0-th, 7-th, 15-th, 23-th, 31-th} into the beginning, middle, and end layer of the Deit architecture. We further compute the corresponding representation similarity for these settings, as shown in Fig. 5.

We found that higher Mutual-KNN scores correlate with greater performance gains from inserting LLM blocks, especially when inserted at the end of DEiT. On one hand, higher similarity suggests that the model learns more general features from the frozen LLM blocks, enhancing generalization. On the other hand, as shown in Fig. 3, insertation at the end yields gradient gains for all ViT blocks.

Auxiliary Training Method

In the previous section, we show that frozen pretrained LLM blocks act as gradient rectifiers to enhance training generalization, as indicated by higher GSNR. The further back

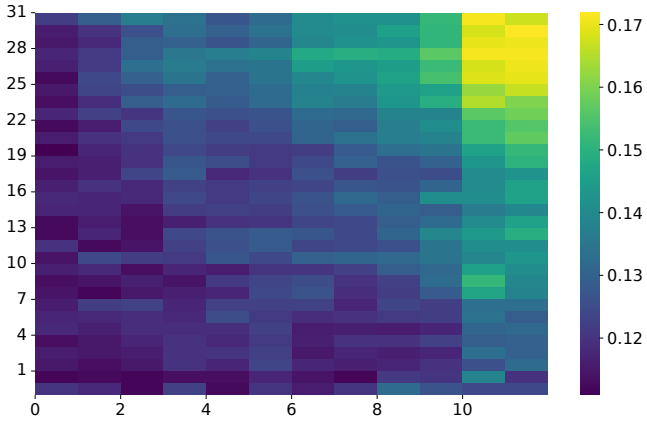

Figure 4: Mutual-KNN scores between the 32 blocks of pre-trained LLaMA-7B and the 12 blocks of pre-trained ViT-S. In both vision and linguistic models, higher layers exhibit higher similarity, indicating that they learn representations that are more general across different modalities.

the LLM block is inserted, the more ViT parameters are affected. Additionally, as the representations become more similar, the performance gains increase.

However, integrating a large LLM block into a small model to enhance downstream task performance incurs additional costs, increasing the parameter count from 22.05M to 227.58M and reducing inference speed from 75 samples per second to 70 samples per second.

Inspired by Tiwari and Shenoy (2023); Foret et al. (2020), we aim to enhance gradient coherence by using the gradient path through the LLM block as an auxiliary task, while employing only the pure ViT architecture during inference.

In the training process as described in Fig. 6, we set up two gradient backpropagation pathways, they share nearly all of the trainable parameters (i.e., all of the ViT blocks).

Insert Pos	0	7	15	23	31
Early	68.12	67.54	67.78	67.40	68.73
Middle	69.01	70.22	73.18	73.50	73.76
End	70.65	71.39	71.95	74.28	74.14

Table 3: Inserting different LLaMA blocks into various positions within the DEiT framework results in significant performance variations.

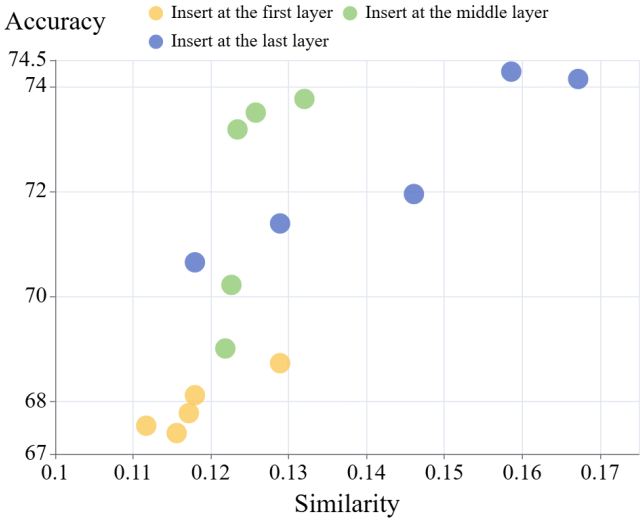

Figure 5: The greater the similarity between the LLM block and the nearby ViT blocks at the insertion position, the larger the performance gain achieved. This effect is most pronounced when the LLM block is inserted after the last ViT layer, due to the highest similarity and gradient rectification effect benefiting all parameters.

In the naive pathway, in order to preserve the effectiveness of ViT itself in visual tasks, we pass the data flow through ViT and use it directly for downstream tasks. In the additional pathway, the data flow not only goes through the vanilla ViT, but also through the LLaMA block that we have inserted, introducing a rectification effect of gradient consistency in this gradient backpropagation pathway, thereby improving the overall consistency of gradient updates.

In inference process, we retain only the native path, i.e., the pure ViT architecture, and discard the LLM block and its corresponding adapter linear layer. This ensures lower memory and time consumption during inference.

Specifically, assuming LLM block as gradient rectifier, we have two branches as

$$\begin{aligned} &\text{Main}: \text{ViT\_Blocks}(x) \cdot \text{Head}(x) \rightarrow \tilde{y}_{cls}, \\ &\text{Aux}: \text{ViT\_Blocks}(x) \cdot \text{Rectifier} \cdot \text{Head}(x) \rightarrow \tilde{y}_{aux}. \end{aligned} \tag{8}$$

Next, we compute the losses for each branches. Here, $\mathcal{L}_{aux} = \mathcal{L}_{(y,\tilde{y}_{aux})}$ represent the losses after gradient rectification, while $\mathcal{L}_{main} = \mathcal{L}_{(y,\tilde{y}_{cls})}$ corresponds to the results obtains from the pure ViT. And λ represents the weight of

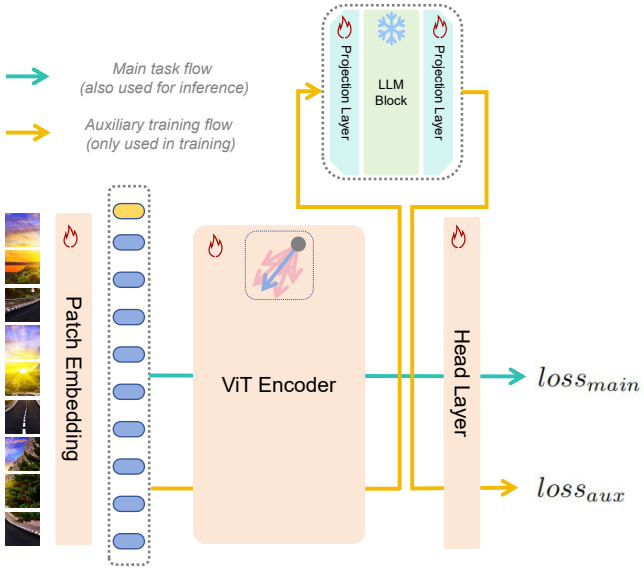

Figure 6: Framework of our auxiliary training methods. During training, gradients propagate through two paths: one that optimizes the downstream task's objective function (blue arrow), and another that enhances gradient coherence via the LLM Block (yellow arrow). In the inference phase, the LLM Block and Adapter layer are removed, with inference relying solely on the pure ViT Path (yellow arrow).

auxiliary tasks, decreasing across training epochs as

$$\lambda = \max(0.10, \frac{\lambda_{\text{init}} \cdot (\text{epoch}_{max} - \text{epoch}_{cur})}{\text{epoch}_{max}}). \tag{9}$$

Due to our primary focus on $\mathcal{L}_{main}$ during inference, in the early stages of training, we set a larger λ to fully leverage the rectifying effect of gradient consistency in $\mathcal{L}_{aux}$. As the training progresses and the model approaches fitting, we increase the weight of $\mathcal{L}_{main}$ to fully train the downstream task capabilities of the vanilla ViT. Finally in Eq. (10), we obtain the final training objective, ensuring optimization of the main task while stabilizing and aligning gradients to capture more primary patterns, thereby enhancing generalization.

$$\mathcal{L}_{total} = (1 - \lambda) \cdot \mathcal{L}_{main} + \lambda \cdot \mathcal{L}_{aux}. \tag{10}$$

Experiments

Datasets

ImageNet-1K (Russakovsky et al. 2015) also known as ILSVRC 2012, is an extensive image dataset structured based on the WordNet hierarchical system. The goal of ImageNet is to offer an average of 1000 images for each synset, with each image being meticulously screened and annotated by humans to ensure quality.

Something-Something-v2 (Goyal et al. 2017) contains 220,847 labeled video clips that depict humans carrying out basic actions with common objects. The dataset aims to enhance the neural models in recognizing subtle hand gestures,

Model	ImageNet-1K						Something-Something-v2		
	ImageNet	ImageNet-C	ImageNet-A	ImageNet-SK	ImageNet-R	GSNR($\times 10^{-5}$)	acc@1	acc@5	GSNR($\times 10^{-3}$)
ViT-S	80.1	57.2	**20.5**	28.9	42.1	6.36	**64.71**	89.15	1.79
+AUX	**80.3**	**57.6**	20.0	**30.1**	**42.3**	**7.10**	64.58	**89.54**	**1.80**
ViT-B	80.6	60.5	23.4	31.9	43.5	8.12	64.97	89.50	0.94
+AUX	**80.9**	**61.1**	**24.2**	**32.3**	**44.3**	**9.34**	**65.85**	**90.08**	**1.02**

Table 4: Performance on ImageNet-1K and SSv2. With auxiliary training, our method achieves the improved performance among image classification and video understand tasks.

Model	CIFAR-100			Caltech 256		
	acc@1	acc@5	GSNR($\times 10^{-5}$)	acc@1	acc@5	GSNR($\times 10^{-5}$)
ViT-T	70.21	90.97	1.76	39.65	63.77	1.66
+Aux	**71.27**	**91.06**	**2.11**	**40.26**	**64.12**	**2.37**
ViT-S	71.62	91.35	1.28	47.16	68.10	1.20
+Aux	**73.37**	**91.43**	**1.36**	**48.15**	**69.25**	**1.97**

Table 5: Performance on other datasets. With auxiliary training, our method achieves the improved performance among several image classification tasks.

λ	CIFAR-100		Caltech 256		BAR	
	acc@1	acc@5	acc@1	acc@5	acc@1	acc@3
0	71.62	91.35	47.16	68.10	33.49	58.56
0.15	72.88	91.35	47.26	68.67	34.70	69.57
0.25	72.92	91.37	47.12	68.69	34.55	68.04
0.35	**73.37**	91.43	**48.15**	**69.25**	**34.86**	**68.20**
0.45	72.75	**91.47**	47.61	69.02	34.63	67.58

Table 6: Ablation study on the auxiliary loss weight λ.

such as placing an item into another object, flipping something over, and covering an object with another item.

CIFAR-100 (Krizhevsky et al. 2009), one of the most classic datasets in computer vision, consists of 100 categories with sixty thousand images.

Caltech-256 (Griffin, Holub, and Perona 2007) contains 256 different objects for generic object recognition, with at least 80 images per category. We split the dataset into training and testing sets in a 7:3 ratio.

Implementation Details

For the video understanding task, we use VideoMAE (Tong et al. 2022) and train it on the SSV2 dataset. we train for 40 epochs with a batch size set to 24 for ViT-S, and 30 epochs with a batch size of 12 for ViT-B. Note that, due to computational constraints, we fine-tune the model based on pretrained weights rather than training from scratch.

For image classification task, we utilize DEiT (Touvron et al. 2021) and train for 300 epochs. For ImageNet, we set batch size to 1024. For Cifar-100 and Caltech-256, we set batch size to 256. And for Bar, we set batch size to 64.

We use the AdamW optimizer with a learning rate set to $5e$-4, weight decay set to $1e$-5 and cosine annealing scheduling for updates on A800 and RTX 3090 device.

Performance Evaluation

After incorporating auxiliary training, we compare its performance with that of the vanilla ViT on ImageNet and SSv2 dataset in terms of accuracy. For each dataset, we calculate its $gsnr$ during the training process. As shown in Tab. 4, the $gsnr$ typically increases by over 10%, with a concurrent improvement in ImageNet, which also means a positive correlation between gradient coherence and model performance.

It's noteworthy that $acc@1$ performance on the SSv2 isn't remarkable due to it is the result of fine-tuning the pretrained weights from the SSv2 dataset, and since the model has already learned sufficient representations through pretraining, the boost brought by the LLaMA block is limited, resulting in a slight decreased performance.

Additionally, as shown in Tab. 5, we also conduct experiments on other datasets, further validating the effectiveness of auxiliary training, which demonstrates performance gains for both ViT-T and ViT-S models.

Ablation Studies

In Tab. 6, we focus on the impact of the weight of auxiliary training on overall performance. We adjust the weight λ and observe that with a higher weight of auxiliary loss, the model exhibits better performance in both top-1 and top-5 accuracy metrics. However, when the rectification effect of gradients becomes too strong, it can impair model performance. For example, at $\lambda = 0.45$, there is a decrease of 0.62% in $acc@1$ compared to $\lambda = 0.35$ in CIFAR-100. This is attributed to the noise gradients introduced by the auxiliary loss, which hinders the optimization convergence of the main loss associated with classification tasks.

Conclusion

In this work, we analyze the potential and mechanisms of LLMs as general task performance enhancers. From the perspective of gradient dynamics, we show that frozen pretrained LLMs exhibit a gradient rectification effect, which improves gradient consistency during training and thereby enhances generalization performance.

Furthermore, we explore the relationship between this gain and representation similarity, and propose a strategy for selecting the optimal combination of LLaMA blocks and their insertion positions in the ViT.

Finally, we propose an auxiliary training strategy that leverages the gradient rectification effect from LLM blocks without introducing additional inference costs. We hope this will assist the community in better exploring the general capabilities of LLMs.

Acknowledgments

This research is supported by National Natural Science Foundation of China (Grant No.62276154), Research Center for Computer Network (Shenzhen) Ministry of Education, the Natural Science Foundation of Guangdong Province (Grant No.2023A1515012914 and 44030024103310080177O), Basic Research Fund of Shenzhen City (Grant No.JCYJ20210324120012033 and GJHZ20240218113603006), the Major Key Project of PCL for Experiments and Applications (PCL2021A06), Alibaba Research Intern Program.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a Visual Language Model for Few-Shot Learning. In *Advances in Neural Information Processing Systems*.

Bardes, A.; Ponce, J.; and LeCun, Y. 2022. VICREG: VARIANCE-INVARIANCE-COVARIANCE REGULARIZATION FOR SELF-SUPERVISED LEARNING. In *10th International Conference on Learning Representations, ICLR 2022*.

Chatterjee, S.; and Zielinski, P. 2022. On the generalization mystery in deep learning. *arXiv preprint arXiv:2203.10036*.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240): 1–113.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Foret, P.; Kleiner, A.; Mobahi, H.; and Neyshabur, B. 2020. Sharpness-aware Minimization for Efficiently Improving Generalization. In *International Conference on Learning Representations*.

Fort, S.; Nowak, P. K.; Jastrzebski, S.; and Narayanan, S. 2019. Stiffness: A new perspective on generalization in neural networks. *arXiv preprint arXiv:1901.09491*.

Goyal, R.; Ebrahimi Kahou, S.; Michalski, V.; Materzynska, J.; Westphal, S.; Kim, H.; Haenel, V.; Fruend, I.; Yianilos, P.; Mueller-Freitag, M.; et al. 2017. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, 5842–5850.

Griffin, G.; Holub, A.; and Perona, P. 2007. Caltech-256 object category dataset.

Hong, Y.; Zhen, H.; Chen, P.; Zheng, S.; Du, Y.; Chen, Z.; and Gan, C. 2023. 3D-LLM: Injecting the 3D World into Large Language Models. *NeurIPS*.

Huh, M.; Cheung, B.; Wang, T.; and Isola, P. 2024. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*.

Jin, M.; Yu, Q.; Huang, J.; Zeng, Q.; Wang, Z.; Hua, W.; Zhao, H.; Mei, K.; Meng, Y.; Ding, K.; et al. 2024. Exploring Concept Depth: How Large Language Models Acquire Knowledge at Different Layers? *arXiv preprint arXiv:2404.07066*.

Kornblith, S.; Norouzi, M.; Lee, H.; and Hinton, G. 2019. Similarity of neural network representations revisited. In *International conference on machine learning*, 3519–3529. PMLR.

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

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Liebel, L.; and Körner, M. 2018. Auxiliary tasks in multitask learning. *arXiv preprint arXiv:1805.06334*.

Liu, J.; Bai, Y.; Jiang, G.; Chen, T.; and Wang, H. 2019. Understanding Why Neural Networks Generalize Well Through GSNR of Parameters. In *International Conference on Learning Representations*.

Merullo, J.; Castricato, L.; Eickhoff, C.; and Pavlick, E. 2022. Linearly Mapping from Image to Text Space. In *The Eleventh International Conference on Learning Representations*.

Michalkiewicz, M.; Faraki, M.; Yu, X.; Chandraker, M.; and Baktashmotlagh, M. 2023. Domain Generalization Guided by Gradient Signal to Noise Ratio of Parameters. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6177–6188.

Pang, Z.; Xie, Z.; Man, Y.; and Wang, Y.-X. 2023. Frozen Transformers in Language Models Are Effective Visual Encoder Layers. In *The Twelfth International Conference on Learning Representations*.

Park, Y.-J.; Wang, H.; Ardeshir, S.; and Azizan, N. 2023. Quantifying Representation Reliability in Self-Supervised Learning Models. *arXiv preprint arXiv:2306.00206*.

Ruder, S. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252.

Sun, Z.; Sun, Y.; Yang, L.; Lu, S.; Mei, J.; Zhao, W.; and Hu, Y. 2023. Unleashing the Power of Gradient Signal-to-Noise Ratio for Zero-Shot NAS. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5763–5773.

Taha, A.; Hanson, A.; Shrivastava, A.; and Davis, L. 2021. SVMax: A Feature Embedding Regularizer. *arXiv preprint arXiv:2103.02770*.

Tiwari, R.; and Shenoy, P. 2023. Overcoming Simplicity Bias in Deep Networks using a Feature Sieve. In *ICML*,

volume 202 of *Proceedings of Machine Learning Research*, 34330–34343. PMLR.

Tong, Z.; Song, Y.; Wang, J.; and Wang, L. 2022. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35: 10078–10093.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, 10347–10357. PMLR.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Workshop, B.; Scao, T. L.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A. S.; Yvon, F.; et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Wu, S.; Fei, H.; Qu, L.; Ji, W.; and Chua, T. 2023. NExT-GPT: Any-to-Any Multimodal LLM. *CoRR*, abs/2309.05519.

Yu, X.; Tang, L.; Rao, Y.; Huang, T.; Zhou, J.; and Lu, J. 2022. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19313–19322.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2021a. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3): 107–115.

Zhang, D.; Li, S.; Zhang, X.; Zhan, J.; Wang, P.; Zhou, Y.; and Qiu, X. 2023. SpeechGPT: Empowering Large Language Models with Intrinsic Cross-Modal Conversational Abilities. arXiv:2305.11000.

Zhang, H.; Li, X.; and Bing, L. 2023. Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding. *arXiv preprint arXiv:2306.02858*.

Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X. V.; et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zhang, Z.; Yang, Y.; Yao, Z.; Yan, Y.; Gonzalez, J. E.; Ramchandran, K.; and Mahoney, M. W. 2021b. Improving semi-supervised federated learning by reducing the gradient diversity of models. In *2021 IEEE International Conference on Big Data (Big Data)*, 1214–1225. IEEE.