Depth as Modulation in Weight-Sharing Transformers

Anonymous authors
Paper under double-blind review

Abstract

Weight-sharing architectures provide an efficient design for Transformers. However, their reliance on a single transformation can limit the model's capacity for iterative representation refinement, a process that requires functional specialization across layers. We address this limitation by representing depth through layer-wise perturbations, creating a path toward models that are both parameter-efficient and performant. Our approach iteratively applies a shared block, and we introduce two distinct strategies to perturb its Multi-Head Self-Attention (MHSA) component with each application: a comprehensive QKOV-LoRA and a more parameter-efficient, QK/OV-circuit. The effectiveness of these strategies is validated on vision and language benchmarks, where our models demonstrate favorable performance against layer-sharing counterparts. Our results suggest that layer-wise perturbing a shared structure is an effective principle for developing capable and efficient Transformers.

1 Introduction

Transformer architectures, now central to progress in language and vision, are increasingly understood as systems that perform iterative refinement of representations (Vaswani et al., 2017; Dosovitskiy et al., 2020). Rather than being a simple stack of identical modules, a deep Transformer exhibits functional evolution across layers. Early layers typically handle local token interactions and basic feature extraction, while deeper layers integrate global context and organize the feature space for downstream tasks (Raghu et al., 2021; Geshkovski et al., 2023a). This perspective implies that the model's effectiveness depends on its ability to apply a sequence of functionally distinct transformations, each building on the previous one. Thus, depth is not merely repetition; it is a critical axis for progressive specialization.

A direct way to reduce the parameter count of deep Transformers is to share parameters across layers. Models such as the Universal Transformer (Dehghani et al., 2018) and ALBERT (Lan et al., 2019) demonstrate that repeatedly applying a shared block can yield compact models with competitive performance. However, this strategy introduces a core tension. While parameter sharing achieves efficiency, it imposes a uniform transformation at all depths, which conflicts with the observed need for functional specialization. A single, fixed operator may struggle to guide the multi-stage process of representation refinement that unshared models learn. Although some approaches add small, fixed modules to a shared core (Zhang et al., 2022; Rajabzadeh et al., 2024), the modification remains constant across depth and does not fully resolve this limitation.

Positioning vs. prior work. Parameter sharing across depth has been explored in the Universal Transformer and ALBERT; in vision, compression methods further leverage weight reuse in conjunction with per-layer transformations or prompts (Zhang et al., 2022; Jia et al., 2022). Our contribution is a specific instantiation that perturbs MHSA with small, depth-indexed operators, contrasted with adapters that are invariant across depth. We target either the projection matrices (QKOV-LoRA) or the composite operators that govern attention computation (QK/OV-circuit). We do not claim general superiority over all adapter families; rather, we evaluate when depth-varying perturbations are beneficial and how they trade off accuracy and parameter cost.

To address the uniformity constraint, we propose the principle of *depth-by-perturbation*. We reconcile the efficiency of parameter sharing with the need for functional diversity by representing depth via layer-wise perturbations of a shared block. Concretely, we maintain a common backbone for memory and computational advantages, and introduce small, learnable, layer-specific modifications at each application. This yields a sequence of closely related yet distinct operators, enabling a smooth evolution of function through depth. From a dynamical systems viewpoint—where deep networks are modeled as discrete-time systems—a standard weight-sharing model corresponds to an autonomous

system with a fixed update rule. In contrast, our approach induces non-autonomous behavior: the update rule is modulated at each time step (i.e., depth), allowing more adaptive representational trajectories. We next formalize depth-by-perturbation as a non-autonomous recursion and specify where the perturbations enter the forward pass. We achieve this by introducing small, layer-indexed parameters that systematically alter the MHSA's function, effectively changing the system's update rule at each step.

We target perturbations to the Multi-Head Self-Attention (MHSA) mechanism, which governs inter-token relationships. Analyses show that attention drives the geometric organization of the feature space, for example by grouping semantically similar tokens into clusters (Geshkovski et al., 2023b; Raghu et al., 2021). This consolidation is particularly important in deeper layers. We therefore modulate the projections within MHSA (Q, K, O, V) to directly influence both the routing pattern of token interactions and the aggregation of information at each layer.

We instantiate this principle with two parameter-efficient mechanisms. The first, *QKOV-LoRA*, applies low-rank adaptations (LoRA) to all four projection matrices (Q, K, O, V), providing comprehensive control. The second, *QK/OV-circuit*, is a parsimonious, circuit-inspired design that perturbs *both* the query–key (QK) and output–value (OV) composite operators at each layer via low-rank updates to the corresponding endpoints of the computation, avoiding separate updates to all four matrices. Both methods employ a single shared backbone; functional specialization is encoded entirely in lightweight, depth-indexed perturbations.

This paper makes three contributions. (1) We introduce *depth-by-perturbation* as a principle for building efficient Transformers that retain depth-wise specialization. (2) We develop two concrete MHSA-based implementations—QKOV-LoRA and QK/OV-circuit—with complementary efficiency profiles. (3) Through experiments on vision and language benchmarks, we show that our approach improves upon standard weight-sharing models, particularly in data-limited settings, suggesting that layer-wise perturbations of a shared structure offer a practical route to capable and efficient Transformers.

The remainder of the paper is organized as follows. We review related work in Section 2, provide background and conceptual motivation in Section 3, detail our architectures in Section 4, present experiments in Section 5, and conclude in Section 6.

2 Related Work

Our approach synthesizes principles from three research areas to create Transformers that are both parameter-efficient and functionally specialized across depth. We first discuss the drive for efficiency through parameter sharing, then review evidence for depth-wise specialization, and finally examine lightweight adaptation methods as a mechanism to reconcile these two objectives.

The Efficiency–Uniformity Trade-off in Parameter Sharing. To mitigate the memory and computational costs of deep Transformers, researchers have explored parameter sharing across layers. Models such as the Universal Transformer (Dehghani et al., 2018) and ALBERT (Lan et al., 2019) showed that repeatedly applying a shared block can produce compact models with strong performance. This principle has been refined through more granular strategies, including sharing only middle layers (Reid et al., 2021) or sharing at the level of attention heads (Cao et al., 2024). However, a core trade-off emerges: efficiency is gained at the cost of imposing a uniform transformation at all depths. This uniformity conflicts with the behavior of standard Transformers, where functional specialization evolves through the layers. Our work retains the shared backbone for efficiency but seeks to overcome the uniformity constraint.

The Case for Depth-wise Functional Specialization. There is substantial evidence that a Transformer's depth enables a sequence of functionally distinct processing stages. In language models, lower layers capture surface and syntactic patterns, while upper layers compose semantic meaning (Tenney et al., 2019; Jawahar et al., 2019; Rogers et al., 2020; Hewitt & Manning, 2019; Ethayarajh, 2019; Clark et al., 2019; Geva et al., 2021). In vision, ViTs exhibit a shift from diverse low-level features to integrated, consolidated representations in later layers; attention maps increasingly highlight object-level structure (Raghu et al., 2021; Caron et al., 2021). This consolidation supports token merging or pruning with little accuracy loss (Bolya et al., 2023; Rao et al., 2021). Together, these findings indicate that depth benefits from functionally distinct transformations rather than a single fixed operator.

Lightweight Adapters as a Mechanism for Specialization. Parameter-efficient tuning methods, originally developed for adapting large models, offer tools to reintroduce depth-wise variation at low cost. Low-Rank Adaptation (LoRA) modifies a pretrained matrix W_0 with a low-rank update $W = W_0 + BA$, training only A, B (Hu et al., 2022). Beyond fine-tuning, this idea can serve as a design element for new architectures. In our framework, layer-indexed adapters systematically vary the function of a shared block at each application, enabling non-autonomous depth dynamics while keeping the backbone fixed. Concretely, we instantiate two MHSA-focused mechanisms: (i) *QKOV-LoRA*, which applies LoRA to all four projections (Q, K, O, V) for comprehensive control; and (ii) *QK/OV-circuit*, a parsimonious, circuit-inspired design that perturbs *both* the query–key (QK) and output–value (OV) composite operators at every layer via low-rank updates, avoiding separate updates to all four projection matrices. This realizes depth-wise specialization with a small parameter budget while preserving the benefits of weight sharing. Building on this motivation, the next chapter formulates two specific methods for selectively perturbing MHSA, drawing inspiration from the separation of QK (information routing) and OV (information aggregation) in Transformer Circuits.

3 Background

Our work lies at the intersection of (i) viewing deep networks as discretized dynamical systems, (ii) empirical evidence for depth-wise functional specialization in Transformers, and (iii) architectural principles for parameter-efficient design. We synthesize these perspectives to motivate representing depth through small, learnable perturbations applied to a shared block.

3.1 Transformers as Discretized Dynamical Systems

A useful framework for understanding deep residual networks is to view them as discretizations of continuous-time ordinary differential equations (ODEs) (Weinan, 2017; Chen et al., 2018). In this view, the forward pass corresponds to numerically integrating a learned vector field over time (depth), with each layer representing a time step. This perspective brings tools from numerical analysis and dynamical systems to study stability, convergence, and expressivity (Haber & Ruthotto, 2017).

For Transformers, this viewpoint is insightful when the token set is modeled as an interacting particle system (Geshkovski et al., 2023a). Let $x_i(t) \in \mathbb{R}^d$ denote the representation of token i at continuous depth t. A generic ODE decomposition separates inter-token interactions (MHSA) and per-token nonlinearity (FFN):

$$\frac{\mathrm{d}x_i(t)}{\mathrm{d}t} = \underbrace{F\big(x_i(t), \{x_j(t)\}_{j=1}^N, t\big)}_{\text{interaction via MHSA (diffusion)}} + \underbrace{G\big(x_i(t), t\big)}_{\text{self-dynamics via FFN (convection)}}. \tag{1}$$

A standard Transformer block can be interpreted as an operator-splitting scheme (e.g., Lie–Trotter) that alternates these two operators (Lu et al., 2019).

The interaction term F can be related to the *Transformer Circuits* decomposition (Elhage et al., 2021), which separates attention into a routing sub-circuit (QK) and a content sub-circuit (OV). Writing a single-head form for clarity,

$$F_i(t) = \sum_{h=1}^{H} \frac{1}{L_{i,h}} \sum_{j=1}^{N} \exp\left(\underbrace{\frac{x_i^\top(t)\, W_Q^{h\top}(t)\, W_K^h(t)\, x_j(t)}{\sqrt{d_k}}}_{\text{QK circuit: information routing}} \right) \underbrace{\big(W_O^h(t)\, W_V^h(t)\, x_j(t)\big)}_{\text{OV circuit: content aggregation}}, \tag{2}$$

where $L_{i,h}$ is the softmax normalizer. Modulating the QK circuit changes the data-dependent interaction topology (routing), while modulating the OV circuit changes how information is transformed and aggregated. This separation provides two complementary levers for architectural control.

3.2 Empirical Evidence for Depth-wise Functional Specialization

A large body of work shows that layers in deep Transformers are not redundant but form a pipeline of related yet distinct computations.

Language. Probing studies on BERT-like models show that lower layers encode surface and syntactic information, while higher layers compose more abstract semantics (Tenney et al., 2019; Jawahar et al., 2019; Rogers et al., 2020). The "structural probe" reveals that syntactic structure is linearly recoverable from middle layers (Hewitt & Manning, 2019). Contextualization increases with depth, with upper layers exhibiting lower self-similarity and higher anisotropy (Ethayarajh, 2019). Attention heads specialize in roles such as delimiter detection or dependency tracking (Clark et al., 2019), and FFNs store abstract patterns (Geva et al., 2021).

Vision. In ViTs, early layers extract diverse low-level features, while later layers consolidate and integrate information (Raghu et al., 2021). Self-supervised ViTs (e.g., DINO) show that later attention heads often highlight object-level structure (Caron et al., 2021). Consolidation implies redundancy among tokens, enabling merging or pruning in late layers with minimal accuracy loss (Bolya et al., 2023; Rao et al., 2021).

Together, these findings suggest that depth is best realized as a sequence of related but distinct transformations, rather than repeated applications of a single fixed operator.

3.3 Late-Layer Dynamics: Refinement, Redundancy, and Residual Streams

Residual connections propagate features forward, creating a stable residual stream (Raghu et al., 2021). Update magnitudes typically shrink with depth, and adjacent late layers often show high representation similarity (e.g., via CKA) (Kornblith et al., 2019). While this has been interpreted as redundancy—e.g., many heads or layers can be pruned with limited impact (Voita et al., 2019; Michel et al., 2019; Dalvi et al., 2020)—a more accurate view is that late layers perform fine-grained refinement. Adaptive computation mechanisms (e.g., ACT in Universal Transformers) support this, allocating more steps to harder inputs (Dehghani et al., 2018; Xin et al., 2020). Vision architectures like CaiT explicitly specialize late blocks (e.g., class attention) to guide consolidation (Touvron et al., 2021).

3.4 The Uniformity Constraint of Parameter Sharing and the Case for Modulation

Layer-wise parameter sharing is an effective route to memory- and compute-efficient Transformers, as in the Universal Transformer and ALBERT (Dehghani et al., 2018; Lan et al., 2019). However, strict sharing yields an *autonomous* system with a fixed update rule, conflicting with the empirically non-autonomous behavior of standard deep stacks. A single, static operator is ill-suited to cover the progression from early feature extraction to late-stage consolidation.

This motivates a *quasi-autonomous* design: retain a shared backbone for efficiency while introducing small, learnable, depth-indexed perturbations that modulate the update rule. Targeting MHSA—the interaction term governing inter-token geometry—provides direct control over representational dynamics. In subsequent sections we instantiate this via two MHSA-focused mechanisms: (i) *QKOV-LoRA*, which applies LoRA to all four projection matrices (Q, K, O, V); and (ii) *QK/OV-circuit*, which perturbs *both* the QK (routing) and OV (content) composite operators at every layer using low-rank updates, achieving depth-wise specialization with a compact parameter budget while preserving weight-sharing efficiency.

4 Method: Engineering Non-Autonomous Dynamics in Transformers

To bridge the gap between the efficiency of parameter sharing and the necessity of depth-wise functional specialization, we propose a concrete methodology founded on the *depth-by-perturbation* principle. This section translates the theoretical concept of a non-autonomous dynamical system into a practical architectural framework. We detail our approach to systematically introduce learnable, layer-indexed variations into a shared computational block, thereby enabling the iterative refinement that is characteristic of high-performing deep Transformers.

4.1 The General Framework: Perturbing a Shared Core

Our methodology is centered on simulating a deep, L-layer Transformer by iteratively applying a single, pretrained encoder block, denoted $\mathcal{B}_{\text{shared}}$. The central innovation lies in modulating this block's behavior at each step of the iteration. The state transition for token representations h_i at effective depth i is formalized as:

$$h_{i+1} = \mathcal{B}_{\text{shared}}(h_i; \theta_{\text{frozen}}, \delta_i), \quad \text{for } i \in \{l, \ldots, L\}, \tag{3}$$

where h_i denotes the representation at effective depth i. Here, θ_{frozen} comprises the vast majority of the block's parameters, inherited from a pretrained model and kept frozen. This strategy is crucial for anchoring the model in a robust, pre-established feature space and preventing catastrophic forgetting. The perturbation is encapsulated in δ_i, a small, fully trainable module whose parameters are unique to each effective depth i. This formulation creates a *quasi-autonomous* system—one that retains a stable, shared core while exhibiting adaptive, depth-variant dynamics. Here, δ_i acts only within MHSA. It can (i) modify the per-projection weights via low-rank updates (QKOV-LoRA) or (ii) add low-rank corrections at the composite sites used to form attention logits and the OV contraction (QK/OV-circuit). We keep FFN frozen to attribute changes to inter-token interactions.

The choice of where to apply this perturbation is a critical design decision, guided by the functional roles of the Transformer's sub-components. We exclusively target the Multi-Head Self-Attention (MHSA) module. As indicated by the dynamical systems perspective (Section 3.1), MHSA functions as the interaction term, governing the flow of information between tokens. Modulating MHSA allows us to directly control the evolution of the system's geometry—that is, how token representations are organized relative to one another. In contrast, the Feed-Forward Network (FFN), which acts as a per-token feature transformation (the convection term), remains completely frozen. Fixing the FFN provides a stable, nonlinear basis upon which the attention mechanism can operate, ensuring that changes in dynamics are attributable to controlled modifications of inter-token relationships, rather than concurrent, potentially destabilizing changes in the feature space itself.

4.2 Architectural Instantiations Tailored to Domain

We instantiate this framework through two distinct architectural blueprints, each tailored to the structural properties and empirical regularities of models used in NLP and computer vision.

4.2.1 Injecting Non-Autonomy into Recurrent Language Models

For NLP, our design adapts the ALBERT architecture, a model that exemplifies the strengths and weaknesses of a purely autonomous, recurrent system. Its single, shared encoder block offers extreme parameter efficiency but imposes a functionally static transformation across all layers. Our modification, shown in Figure 1, is designed to alleviate this rigidity. Starting with a pretrained ALBERT-base-v2, we maintain its recurrent application of the shared block. However, we partition its effective depth. For the initial l steps, the model's operation is unchanged, allowing it to perform the foundational syntactic and semantic analysis for which it was trained.

From effective depth $l + 1$ onwards, we introduce our depth-specific MHSA perturbations, δ_i. This intervention is motivated by extensive probing studies (Section 3.2) showing that deeper layers in language models are responsible for higher-level semantic integration and contextual refinement. By enabling the transformation rule to vary in these crucial later stages, we empower the model to learn a more sophisticated and context-sensitive processing pipeline, thereby transforming it from a simple autonomous system into a more powerful non-autonomous one without sacrificing its inherent efficiency.

4.2.2 Imposing Recurrence on Vision Transformers

In contrast to ALBERT, standard Vision Transformers (ViTs) are deep, non-recurrent stacks of unique layers. Directly applying a single shared block from layer 1 would be suboptimal, as early and late layers perform vastly different functions (feature extraction vs. integration). Our design is guided by the empirical observation of late-layer representational similarity (Section 3.3). CKA analyses suggest that while early layers are highly specialized, later layers become more functionally redundant as they iteratively refine a global representation.

Based on this insight, we propose the hybrid architecture in Figure 2. We take a pretrained ViT-B/16 and freeze its first l layers, preserving their role as expert low-level feature detectors. The $l + 1$-th encoder block is then selected to become the recurrent core, $\mathcal{B}_{\text{shared}}$. This block is applied iteratively to replace the original layers from $l + 1$ to L. At each of these recurrent steps, its MHSA module is modulated by a unique perturbation δ_i. This strategy is a principled compromise: it leverages the learned specialization of early layers while replacing the less differentiated later layers with a single, parameter-efficient block. The added perturbations ensure this shared block varies across applications and can learn a sequence of refinement operations appropriate for late-stage feature consolidation.

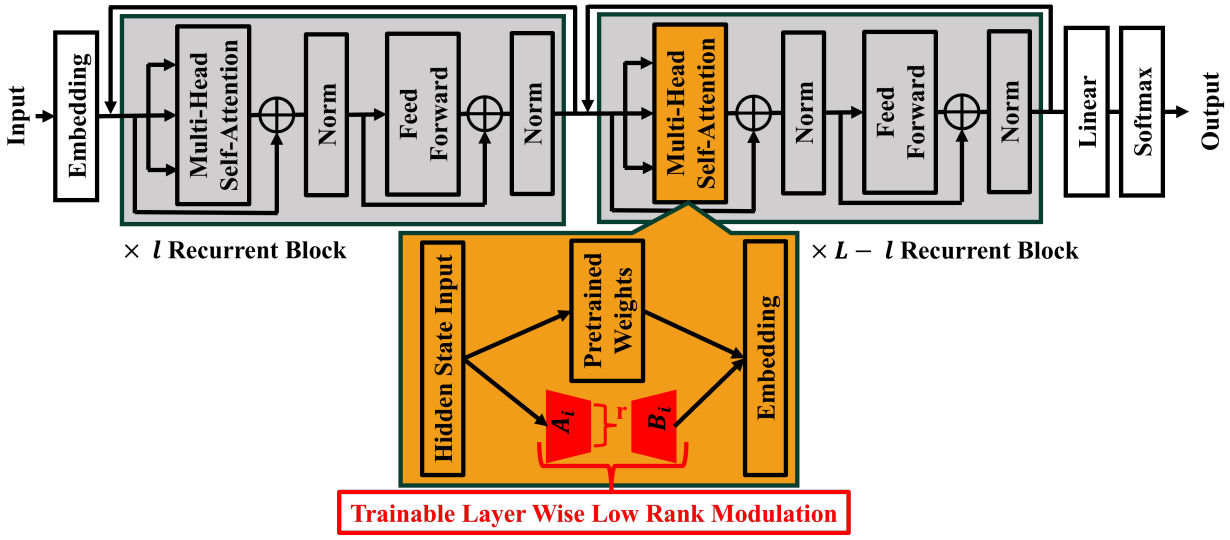

Figure 1: **Depth-by-Perturbation in a Recurrent NLP Model.** We enhance the ALBERT architecture by introducing a unique, learnable perturbation (A_i, B_i) to the MHSA module at each recurrent step $i \geq l + 1$. This intervention transforms the strictly autonomous dynamics of the original model into a non-autonomous system, enabling the capacity for depth-wise specialization.

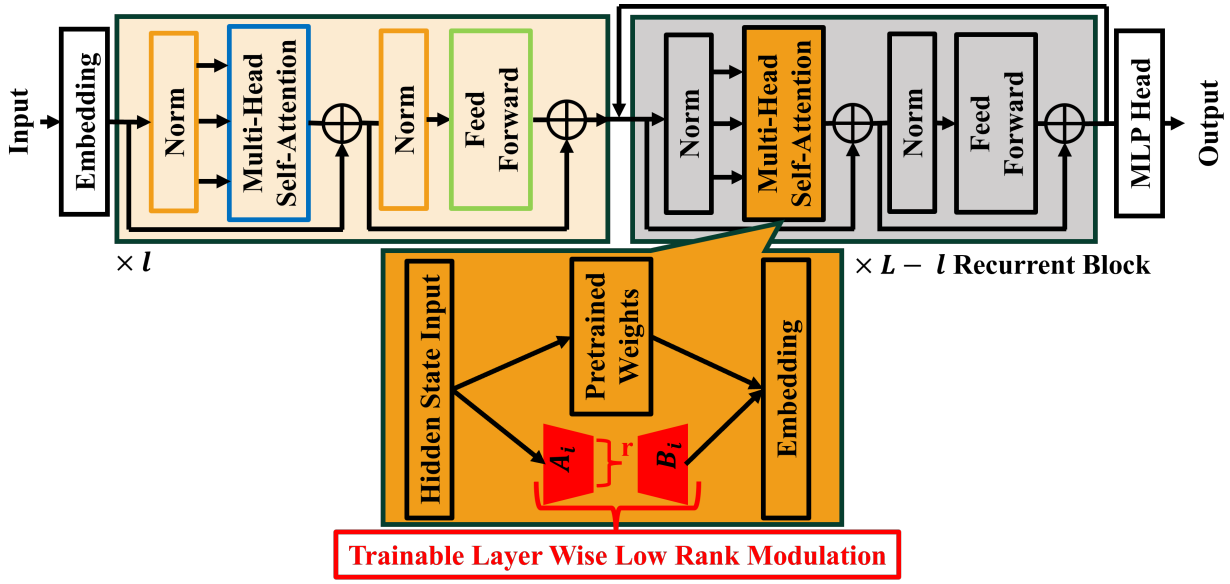

Figure 2: **Inducing a Recurrent, Perturbed Structure in a ViT.** Our ViT architecture is a hybrid design. The specialized initial l layers of a pretrained ViT are preserved. The $l + 1$-th block is repurposed as a recurrent core for the remaining layers, with its MHSA module perturbed (A_i, B_i) at each step. This enables efficient yet functionally evolving processing for high-level feature integration.

4.3 Low-Rank Mechanisms for Efficient Modulation

The perturbations δ_i are implemented efficiently using Low-Rank Adaptation (LoRA). This technique modifies a pretrained weight matrix W_0 with a low-rank update, $W = W_0 + BA$, where only the small matrices A and B are trainable. The low rank r makes this approach highly parameter-efficient. It is particularly well-suited for our purpose as it allows significant functional modification while minimizing the risk of disrupting the valuable knowledge stored in the pretrained weights W_0. We investigate two LoRA-based strategies. For a weight $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, the update uses $A \in \mathbb{R}^{r \times d_{\text{in}}}$ and $B \in \mathbb{R}^{d_{\text{out}} \times r}$. Unless stated otherwise, the rank r is shared across heads and projections.

4.3.1 QKOV-LoRA: A Strategy of Comprehensive Control

This first strategy, QKOV-LoRA, aims for maximum flexibility by applying independent LoRA modules to all four projection matrices (W_Q, W_K, W_O, W_V) of the MHSA. For each effective layer $i \geq l + 1$, this yields four distinct updates:

$$W_{Q,i} = W_{Q,\text{shared}} + B_{Q,i}A_{Q,i}, \qquad W_{K,i} = W_{K,\text{shared}} + B_{K,i}A_{K,i} \tag{4}$$

$$W_{V,i} = W_{V,\text{shared}} + B_{V,i}A_{V,i}, \qquad W_{O,i} = W_{O,\text{shared}} + B_{O,i}A_{O,i} \tag{5}$$

This comprehensive intervention provides four independent levers to control the attention process at each depth. It allows the model to adjust both information routing (via Q and K) and information processing/aggregation (via O and V), offering a fine-grained tool for sculpting representational dynamics. *Forward sites.* The attention logits use $Q_iK_i^\top/\sqrt{d_k}$ with $Q_i = XW_{Q,i}$ and $K_i = XW_{K,i}$; the value path uses O_iV_i with $V_i = XW_{V,i}$ and $O_i = XW_{O,i}$. *Complexity.* Each projection adds $O(rd)$ parameters and no extra activations beyond the LoRA factors.

4.3.2 QK/OV-circuit: A Strategy of Surgical Intervention

Our second strategy is a more targeted and parsimonious approach inspired by the Transformer Circuits decomposition (Section 3.1). Instead of four independent modifications, this strategy makes two surgical interventions that align with the two primary functions of attention. *At every layer* $i \geq l + 1$, it perturbs *both* the query–key (QK) circuit (information routing) and the output–value (OV) circuit (content aggregation) by applying a single low-rank update to each corresponding composite operator:

$$(W_{Q,i})^\top W_{K,i} \approx (W_{Q,\text{shared}})^\top W_{K,\text{shared}} + B_{QK,i}A_{QK,i}. \tag{6}$$

$$W_{O,i}W_{V,i} \approx W_{O,\text{shared}}W_{V,\text{shared}} + B_{OV,i}A_{OV,i}. \tag{7}$$

By focusing the trainable parameters on these two endpoints of the computation, the method controls the most influential aspects of attention dynamics with a significantly smaller parameter budget, achieving depth-wise specialization while preserving the efficiency benefits of weight sharing. *Implementation.* We insert low-rank corrections directly at (i) the logit site, $S_i \leftarrow S_i + \Delta S_i$ with $\Delta S_i = \hat{B}_{QK,i}\hat{A}_{QK,i}$, and (ii) the OV contraction, $(OV)_i \leftarrow (OV)_i + \Delta(OV)_i$ with $\Delta(OV)_i = \hat{B}_{OV,i}\hat{A}_{OV,i}$. This does not imply a unique pair $(W_{Q,i}, W_{K,i})$ that realizes the same composite; we therefore treat ΔS_i and $\Delta(OV)_i$ as independent, trainable low-rank modules with gradients applied at their insertion sites. *Shapes.* $\hat{A}_{QK,i} \in \mathbb{R}^{r \times n}$ and $\hat{B}_{QK,i} \in \mathbb{R}^{n \times r}$ for n tokens per head; analogous shapes apply to OV after projection. *Cost.* Parameters scale with r and head count but avoid four separate projection updates.

5 Experiments

We conducted a comprehensive series of experiments to rigorously validate the *depth-by-perturbation* principle and to evaluate the efficacy of our proposed architectural instantiations, QKOV-LoRA and QK/OV-circuit. Our empirical investigation is structured to systematically answer three primary research questions that underpin this work: (1) Does our core principle of introducing layer-wise, learnable perturbations effectively overcome the inherent limitations of parameter sharing without depth-wise variation, and does this benefit generalize across different data modalities, namely language and vision? (2) What are the relative merits and trade-offs of our two proposed modulation strategies, the comprehensive QKOV-LoRA and the parsimonious QK/OV-circuit, in terms of performance, parameter efficiency, and suitability for diverse tasks? (3) How robust is our approach in data-scarce scenarios, a critical stress test for any method claiming to improve model efficiency and a key indicator of its practical utility?

The experimental narrative proceeds logically. We first present results in natural language understanding, where we apply our methods to an inherently recurrent, weight-sharing model. We then shift to computer vision, starting with a foundational motivational analysis that empirically justifies our focus on the MHSA module. Subsequently, we conduct large-scale evaluations on ImageNet, including in a challenging low-data regime, and conclude with an analysis of downstream task performance to assess the transferability of the learned representations.

5.1 Experimental Setup

Benchmarks and Models. Our evaluation spans diverse domains to demonstrate generality. For natural language understanding, we selected the SuperGLUE benchmark (Wang et al., 2019), a collection of challenging tasks that require nuanced reasoning. Our base model is a pretrained ALBERT-base-v2 (Lan et al., 2019). The choice of ALBERT is deliberate, as its architecture is built upon strict weight sharing—precisely the regime we aim to improve via depth-wise perturbations. For computer vision, we utilize ImageNet-1k (Deng et al., 2009) for primary evaluation and CIFAR-10 (Krizhevsky et al., 2009) to assess downstream transfer. The foundational model for vision is a pretrained ViT-B/16 (Dosovitskiy et al., 2020).

Models Under Comparison. Our primary models are **QKOV-LoRA** and **QK/OV-circuit**. In NLP, we compare against a strong **LoRA** fine-tuning baseline applied to ALBERT, to separate the benefits of *depth-wise, layer-indexed* perturbations from the general effect of adding trainable parameters. In vision, we include baselines that reuse a shared ViT block across layers (simple weight sharing), as well as variants where LoRA adapters are attached to the shared block dentically for all repeated applications (i.e., the adapter does not vary with depth). These baselines test whether varying the operator across depth is necessary beyond making the shared block learnable.

Implementation Details. To ensure fair and reproducible comparisons, all experiments use AdamW (Loshchilov & Hutter, 2017) with a cosine learning-rate schedule. For SuperGLUE, where variance can be substantial, we report mean and standard deviation over 10 random seeds. For vision experiments, key results are averaged over multiple seeds as noted. Unless otherwise specified, we set LoRA rank $r{=}8$ for language and $r{=}16$ for vision, balancing expressive capacity with parameter efficiency. Full hyperparameters, learning rates, and training configurations are provided in the Appendix. Source code will be released upon publication.

5.2 Results on Natural Language Understanding

Table 1: SuperGLUE development set results. We report mean $\pm$ standard deviation over 10 random seeds. Values in parentheses denote absolute improvement over the LoRA baseline.

Task	BoolQ	CB	COPA	MultiRC	RTE	WiC	WSC	Ave.
LoRA ($r = 16$)	70.1±1.0	66.7±0.8	61.7±4.6	77.4±0.4	70.6±2.5	70.5±1.2	58.0±3.3	67.9
QKOV-LoRA ($l = 4$)	72.8±0.6 *(+2.7)*	78.0±2.2 *(+11.3)*	64.7±2.1 *(+3.0)*	75.7±0.4 *(-1.7)*	70.8±1.6 *(+0.1)*	68.3±0.7 *(-2.2)*	50.3±3.9 *(-7.7)*	68.7 *(+0.8)*
QKOV-LoRA ($l = 8$)	72.6±0.1 *(+2.5)*	81.0±2.2 *(+14.3)*	63.0±1.6 *(+1.3)*	74.3±0.5 *(-3.1)*	72.0±1.5 *(+1.3)*	66.1±1.1 *(-4.4)*	53.8±2.8 *(-4.2)*	69.0 *(+1.1)*
QK/OV-circuit ($l = 4$)	71.0±0.1 *(+0.9)*	72.0±1.7 *(+5.4)*	61.7±2.6 *(0.0)*	76.4±0.5 *(-1.0)*	68.4±2.1 *(-2.3)*	65.8±1.1 *(-4.8)*	51.6±3.7 *(-6.4)*	66.7 *(-1.2)*
QK/OV-circuit ($l = 8$)	70.2±0.7 *(+0.2)*	73.8±3.7 *(+7.1)*	65.7±3.3 *(+4.0)*	73.3±0.3 *(-4.1)*	72.8±2.5 *(+2.2)*	67.7±1.5 *(-2.9)*	57.1±1.8 *(-1.0)*	68.6 *(+0.8)*

In our natural language understanding experiments, we evaluated the principle of depth-by-perturbation by applying our methods to ALBERT-base-v2, a model architecturally defined by strict parameter sharing. To demonstrate that our proposed method achieves performance gains beyond simple parameter addition, we compared our QKOV-LoRA and QK/OV-circuit models, both using a low rank of $r = 4$, against a formidable baseline: standard LoRA fine-tuning with a significantly larger rank of $r = 16$. This setup was designed to test the hypothesis that the strategic, depth-wise allocation of trainable parameters is more impactful than simply increasing the capacity of a single, static adapter.

The results, presented in Table 1, lend strong support to this hypothesis. Our QKOV-LoRA model, with perturbations commencing at the eighth effective layer ($l = 8$), achieved an average SuperGLUE score of 69.0. This score represents a

+1.1 point improvement over the more heavily parameterized LoRA baseline. The performance gains were not uniform across tasks; they were exceptionally pronounced on benchmarks requiring complex reasoning and semantic integration. For instance, on the CommitmentBank (CB) task, our model showed a remarkable +14.3 point improvement, and on BoolQ, a +2.5 point gain. These results suggest that enabling the model to apply a sequence of distinct transformations is highly beneficial for tasks that involve iterative refinement of representations.

However, the introduction of depth-wise variation also revealed a trade-off. On tasks such as WiC and WSC, our models underperformed the baseline. This outcome indicates that our current implementation, which applies a uniform perturbation schedule across all later layers, may not be optimally suited for tasks that demand highly specialized, and potentially localized, computational patterns. For instance, tasks like CB and BoolQ require iterative refinement of semantic relationships across entire sentences, where our layer-wise variable transformation rules proved particularly effective. In contrast, pronoun resolution tasks like WSC rely heavily on highly localized and specialized relationships between specific tokens, suggesting that our relatively smooth perturbation schedule may have interfered with the model's native specialized computational patterns. The superior performance of the $l = 8$ configuration over the $l = 4$ one also suggests that allowing the model's foundational layers to operate as originally trained, while introducing perturbations in later, more abstract processing stages, is a more effective strategy.

Our more parsimonious QK/OV-circuit method further validates the core principle. Despite its more constrained design, the $l = 8$ configuration also surpassed the LoRA baseline, achieving an average score of 68.6. This demonstrates that the benefits of depth-by-perturbation are robust and not merely an artifact of the comprehensive control offered by QKOV-LoRA. By directly modulating the core QK (routing) and OV (content) computations, this approach efficiently instills functional diversity across layers. Collectively, these findings indicate that transforming an autonomous weight-sharing model into a non-autonomous one via depth-indexed perturbations is a parameter-efficient and effective strategy for enhancing performance on complex language tasks.

5.3 Results on Computer Vision

5.3.1 Motivational Analysis: The Critical Role of MHSA

A core premise of our approach is that MHSA, which governs inter-token interactions, is the most important component to modulate across depth, while FFN is more amenable to sharing. To test this directly, we construct three ViT variants with recurrence beginning at a specified *Share Index* l: (i) sharing the **Full** block, (ii) sharing only **MHSA** (unique FFNs), and (iii) sharing only **FFN** (unique MHSAs).

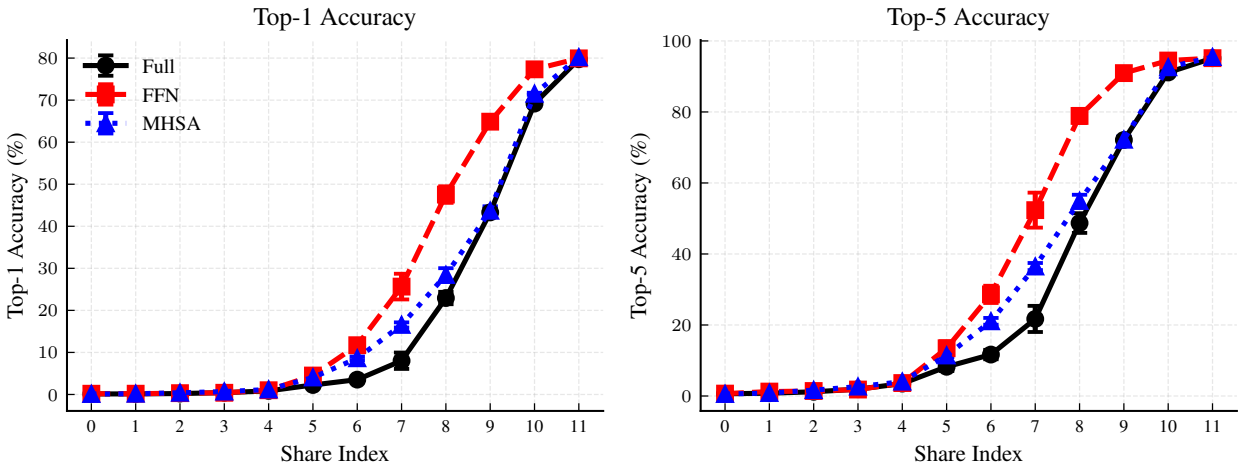

Figure 3: **Impact of component-wise parameter sharing in ViT-B/16 on ImageNet.** Performance vs. "Share Index" (the layer from which components are reused). Sharing MHSA causes a substantially larger degradation in Top-1/Top-5 accuracy than sharing FFN, empirically justifying MHSA-focused perturbations. Top-1/Top-5 are single-crop, center evaluation; resolution matches the base checkpoint.

As shown in Figure 3, sharing MHSA leads to a severe performance collapse compared to sharing FFN. For example, when sharing begins at the 8th layer, the FFN-sharing model maintains Top-1 above 40%, whereas the MHSA-sharing model drops below 20%. This stark difference supports the view that MHSA's role is highly depth-specific—early layers tend to establish local relations, while later layers integrate global context and consolidate object-level structure. In contrast, the per-token FFN transformation is more uniform across depth, making it more suitable for sharing. This analysis motivates our decision to perturb MHSA while freezing FFN.

5.3.2 Evaluating Depth-by-Perturbation on ImageNet

Grounded by the motivational analysis, we evaluate **QKOV-LoRA** and **QK/OV-circuit** on ImageNet-1k against baselines that either (a) reuse a shared block across layers with no variation, or (b) attach LoRA to the shared block but keep the adapter identical at all repeated applications (i.e., no depth-wise change). These baselines test whether making the shared operator learnable is sufficient, or whether it must vary with depth. The purpose of this experiment is not to simply match the number of parameters, but rather to validate the inherent importance of "depth-dependent variation" itself.

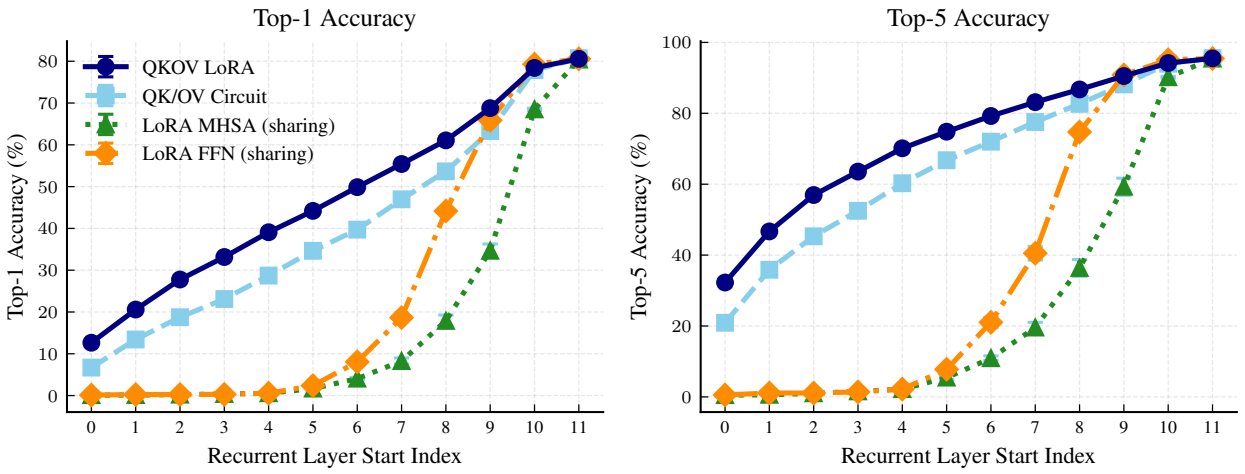

Figure 4: **QKOV-LoRA and QK/OV-circuit on ImageNet.** QKOV-LoRA and QK/OV-circuit (r=16) vs. (i) pure sharing and (ii) a shared block with depth-invariant adapters. Depth-varying perturbations recover substantially more accuracy, especially when recurrence starts early.

Figure 4 shows that baselines without depth-wise variation—whether purely shared or equipped with an adapter that is fixed across all repeated applications—recover little accuracy relative to naive sharing. In contrast, both **QKOV-LoRA** and **QK/OV-circuit** attain stronger results when the shared operator varies across depth. We refrain from causal claims; ablations isolating schedule vs. capacity are left to future work. Notably, even when recurrence begins early (e.g., index 2), our methods build an effective deep processing pipeline, surpassing 30% Top-1 accuracy. The trade-off between the recurrent start index l (which controls parameter savings) and final accuracy is smooth, providing a practical knob for balancing model size and performance. These findings indicate that varying the shared operator across depth is key, beyond merely making it trainable.

5.3.3 Performance in a Low-Data Regime

A crucial test for any parameter-efficient method is its performance in data-scarce scenarios, which often amplify the risk of overfitting. We therefore evaluated our approach in a demanding low-data regime, using only 10% of the ImageNet-1k training set. Our methodology is particularly well-suited for this setting; by freezing the vast majority of the pretrained ViT-B/16 backbone and training only the small, depth-indexed perturbation modules, we anchor the learning process to a robust, pre-established feature space. This strategy is designed to focus the model's limited learning capacity on refining its dynamics rather than re-learning foundational features from scratch.

Table 2: Few-shot performance on ImageNet-1K using a ViT-B/16 base model (trained on 10% of the data). "Layer Index" corresponds to the starting layer l from which the shared block is iteratively applied.

Model		Top-1 Acc. (%)	Params(M)
ViT-B/16 (Base)		84.53	86.71
QKOV LoRA	Layer Index 2	18.48 ± 4.60	16.60
	Layer Index 5	40.61 ± 1.58	37.55
	Layer Index 8	57.83 ± 0.20	58.49
	Layer Index 11	79.52 ± 0.10	79.44
QK/OV Circuit	Layer Index 2	21.73 ± 0.21	15.81
	Layer Index 5	41.52 ± 0.07	36.97
	Layer Index 8	57.75 ± 0.17	58.13
	Layer Index 11	79.46 ± 0.03	79.29

As shown in Table 2, the results reveal a clear and tunable trade-off between the recurrence start index l, the resulting parameter savings, and the final Top-1 accuracy. Both of our methods maintain strong performance despite the significant reduction in training data. For instance, with the QK/OV-circuit method, setting the start index $l = 8$ yields a model with approximately 33% fewer parameters than the ViT-B/16 base, yet it retains a respectable 57.75% Top-1 accuracy. This demonstrates that our approach provides practical operating points for balancing model size and performance, a key advantage in real-world applications where data may be limited.

This low-data experiment also provides a valuable large-scale comparison between our two modulation strategies. The QK/OV-circuit consistently demonstrates superior training stability, evidenced by its substantially lower standard deviation across all configurations, particularly at shallower start indices ($l = 2, 5$). Furthermore, it achieves higher mean accuracy in these early-recurrence settings and performs nearly identically to the more comprehensive QKOV-LoRA at deeper start points ($l = 8, 11$). This combination of stability and competitive accuracy suggests that the more constrained, circuit-inspired design may be a more robust and efficient choice when training data is scarce.

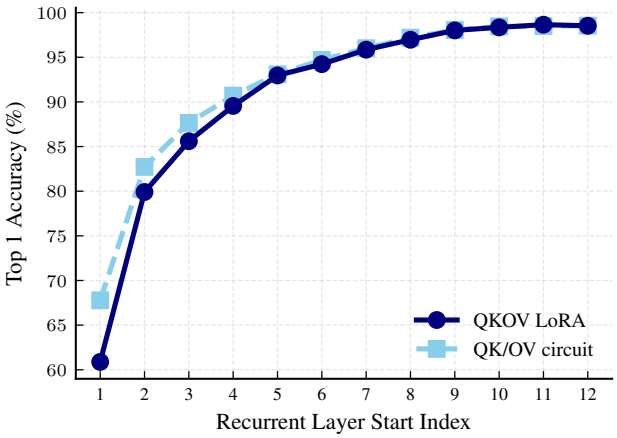

Figure 5: **Downstream performance on CIFAR-10.** QK/OV-circuit is slightly better when recurrence starts earlier, consistent with a regularization effect from perturbing composite QK and OV across layers.

5.3.4 Performance on a Downstream Classification Task

To assess the generalizability of the representations learned by our compressed models, we evaluated their performance on a downstream classification task by fine-tuning them on CIFAR-10. This transfer learning setup serves as a critical test of whether the models have retained core visual features despite significant parameter reduction. The results, illustrated in Figure 5, demonstrate that both the QKOV-LoRA and QK/OV-circuit methods yield models with strong transferability.

As expected, we observe a clear trend across both architectures: performance on CIFAR-10 improves as the recurrence start index l increases. This is consistent with the understanding that preserving more of the original model's specialized early layers, which act as powerful feature extractors, provides a better foundation for adapting to a new task. This creates a predictable trade-off between parameter efficiency and downstream accuracy, allowing for a principled choice of model configuration based on specific application requirements.

A direct comparison of the two strategies reveals a noteworthy pattern. While both methods perform comparably when recurrence begins in later layers (e.g., $l \geq 8$), the QK/OV-circuit consistently shows a slight performance advantage at shallower start indices. This suggests that its more constrained, circuit-focused perturbation may act as a beneficial regularizer, preventing overfitting to the ImageNet pretraining task and fostering the learning of more generalizable features. Consistent with our findings in the low-data regime, this superior performance in challenging (i.e., more heavily compressed) configurations highlights the robustness of the circuit-inspired design.

6 Conclusion

This paper introduced the *depth-by-perturbation* principle, an architectural strategy that reconciles the parameter efficiency of weight-sharing Transformers with the need for depth-wise functional specialization. Grounded in the view of deep networks as discrete-time dynamical systems, a strictly weight-sharing model corresponds to an *autonomous* update rule that is fixed across depth. We argued that such a fixed rule limits the capacity for iterative refinement. Our principle replaces this rigidity with a *non-autonomous* formulation: a single shared encoder block is applied repeatedly, while its Multi-Head Self-Attention (MHSA) is perturbed at each layer by small, learnable modules. This allows a compact model to realize a sequence of related but distinct transformations, approximating the functional evolution observed in deeper, unshared stacks.

We instantiated this idea with two mechanisms that focus on MHSA. **QKOV-LoRA** applies LoRA to all four projection matrices (Q, K, O, V), offering fine-grained control of both information routing and content aggregation at each layer. **QK/OV-circuit** takes a more parsimonious route by perturbing *both* composite operators that define the attention computation—QK (routing) and OV (aggregation)—at every layer via low-rank updates, avoiding separate updates to all four projection matrices. In both cases, the Feed-Forward Network (FFN) remains fixed, serving as a stable nonlinear basis while MHSA governs layer-wise modulation of inter-token geometry.

Our experiments across language and vision support the effectiveness and generality of this principle. On natural language understanding, retrofitting ALBERT with depth-indexed perturbations improved performance over a strong LoRA baseline—notably, with a smaller parameter budget. The largest gains occurred on tasks such as CB and BoolQ that benefit from multi-step semantic refinement, suggesting that how parameters are added—namely, to vary the transformation with depth—matters more than simply increasing capacity. On computer vision benchmarks, naive weight sharing of ViT layers degraded accuracy, particularly when sharing MHSA. In contrast, our methods constructed compact models that preserved high ImageNet accuracy by varying the shared operator across layers. We further observed robustness in a low-data regime (10% of ImageNet-1k), where training only the small perturbation modules anchored optimization to a strong pretrained backbone and reduced overfitting. Finally, both methods transferred well to CIFAR-10, with QK/OV-circuit slightly ahead when recurrence started earlier, consistent with a beneficial constraint from targeting composite operators.

Limitations. While our uniform perturbation schedule proved effective on average, its fixed nature may not be optimal for all tasks, as suggested by the results on WiC/WSC. This highlights a promising direction for future work: developing adaptive or input-conditional perturbation schedules that can allocate functional diversity more dynamically. We also freeze FFN, which may limit adaptation in some regimes. Future work includes adaptive or input-conditional schedules, selective layer gating, and extending perturbations beyond MHSA when justified by task demands.

In summary, *depth-by-perturbation* offers a practical route to capable and efficient Transformers: maintain a shared backbone for memory and compute efficiency, and encode depth-wise specialization through lightweight, layer-indexed perturbations of MHSA. An additional direction is to develop a more rigorous theoretical account—e.g., representation-similarity analysis over depth, spectral characterizations of the QK and OV operators, or information-theoretic measures—to compare representational trajectories induced by non-autonomous updates against those of standard deep stacks, yielding predictive guidance for ranks, layer ranges, and circuit targets.

References

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=g2mupJmI4pC.

Zouying Cao, Yifei Yang, and Hai Zhao. Head-wise shareable attention for large language models. *arXiv preprint arXiv:2402.11819*, 2024.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 276–286, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4828. URL https://www.aclweb.org/anthology/W19-4828.

Fahim Dalvi, Nut Limsopatham, and Yaser Al-Onaizan. Analyzing redundancy in pretrained transformer models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4830–4837, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.398. URL https://www.aclweb.org/anthology/2020.emnlp-main.398.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 55–65, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1006. URL https://www.aclweb.org/anthology/D19-1006.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. The emergence of clusters in self-attention dynamics. *Advances in Neural Information Processing Systems*, 36:57026–57037, 2023a.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. A mathematical perspective on transformers. *arXiv preprint arXiv:2312.10794*, 2023b.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.446. URL https://aclanthology.org/2021.emnlp-main.446.

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

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4129–4138, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://www.aclweb.org/anthology/N19-1419.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3651–3657, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1356. URL https://www.aclweb.org/anthology/P19-1356.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pp. 709–727. Springer, 2022.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3528. PMLR, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images.(2009), 2009.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-Yan Liu. Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv preprint arXiv:1906.02762*, 2019.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32, 2019.

A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in neural information processing systems*, 34:12116–12128, 2021.

Hossein Rajabzadeh, Aref Jafari, Aman Sharma, Benyamin Jami, Hyock Ju Kwon, Ali Ghodsi, Boxing Chen, and Mehdi Rezagholizadeh. Echoatt: Attend, copy, then adjust for more efficient large language models. *arXiv preprint arXiv:2409.14595*, 2024.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, and Jie Zhou. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13937–13949, 2021.

Machel Reid, Edison Marrese-Taylor, and Yutaka Matsuo. Subformer: Exploring weight sharing for parameter efficiency in generative transformers. *arXiv preprint arXiv:2101.00234*, 2021.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://www.aclweb.org/anthology/P19-1452.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 32–42, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5998–6008, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1580. URL https://www.aclweb.org/anthology/P19-1580.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2246–2251, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.204. URL https://www.aclweb.org/anthology/2020.acl-main.204.

Jinnian Zhang, Houwen Peng, Kan Wu, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Minivit: Compressing vision transformers with weight multiplexing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12145–12154, 2022.

A Experimental Specifications

This appendix provides a comprehensive overview of the experimental configurations, hyperparameters, and implementation details used in this work, supplementing the descriptions in the main text. Our goal is to ensure full reproducibility of our findings. All experiments were conducted using PyTorch (Paszke, 2019) and the Hugging Face Transformers library (Wolf et al., 2020), executed on NVIDIA RTX 6000 Ada Generation GPUs.

A.1 SuperGLUE Experiments

Base Model and Architecture. The foundation for our language experiments was the ALBERT-base-v2 model (Lan et al., 2019). This model is particularly relevant to our study due to its inherent cross-layer parameter sharing, where a single shared encoder block is applied recurrently 12 times to form the network's 12 effective layers. Its compact design (12M parameters) makes it an ideal testbed for evaluating strategies that aim to overcome the limitations of a uniform transformation rule.

Hyperparameter Configuration. For our proposed methods, QKOV-LoRA and QK/OV-circuit, we used a low-rank adaptation with rank $r = 4$, a scaling factor $\alpha = 8.0$, and a LoRA dropout of 0.05. In contrast, the LoRA baseline was configured with a larger rank of $r = 16$ and $\alpha = 32$ to provide a strong, more heavily parameterized comparison point.

Training schedules were carefully tailored to the characteristics of each SuperGLUE sub-task to ensure stable convergence. All configurations used a gradient accumulation of 2 steps. The specific settings were as follows:

- **BoolQ & WiC:** 24 epochs, batch size 16, max sequence length 256.
- **CB:** 48 epochs, batch size 8, max sequence length 256.
- **COPA:** 72 epochs, batch size 8, max sequence length 160.

Table 3: SuperGLUE development set results. We report mean ± standard deviation over 10 random seeds. Values in parentheses denote absolute improvement over the LoRA baseline.

Task	BoolQ	CB	COPA	MultiRC	RTE	WiC	WSC	Ave.
Full finetuning	77.0±1.0	83.9±3.9	67.7±1.2	78.8±0.2	75.5±1.8	71.0±0.7	59.9±2.8	73.4
LoRA ($r = 16$)	70.1±1.0	66.7±0.8	61.7±4.6	77.4±0.4	70.6±2.5	70.5±1.2	58.0±3.3	67.9
QKOV-LoRA ($l = 0$)	73.1±0.8 *(+3.1)*	74.4±4.5 *(+7.7)*	62.0±1.4 *(+0.3)*	76.2±0.1 *(-1.2)*	70.6±1.0 *(0.0)*	69.3±0.5 *(-1.3)*	44.9±2.0 *(-13.1)*	67.2 *(-0.6)*
QKOV-LoRA ($l = 1$)	72.9±0.5 *(+2.8)*	74.4±2.2 *(+7.7)*	61.0±1.6 *(-0.7)*	76.3±0.8 *(-1.1)*	71.1±0.5 *(+0.5)*	68.8±1.4 *(-1.8)*	49.4±3.0 *(-8.7)*	67.7 *(-0.2)*
QKOV-LoRA ($l = 2$)	73.4±0.3 *(+3.3)*	76.2±3.0 *(+9.5)*	65.3±1.9 *(+3.7)*	75.8±0.3 *(-1.6)*	68.6±1.1 *(-2.0)*	67.3±0.7 *(-3.2)*	48.7±5.5 *(-9.3)*	67.9 *(+0.1)*
QKOV-LoRA ($l = 3$)	72.3±0.4 *(+2.3)*	77.4±5.9 *(+10.7)*	64.0±2.4 *(+2.3)*	74.0±2.8 *(-3.4)*	72.6±3.1 *(+2.0)*	69.3±1.4 *(-1.2)*	51.9±0.8 *(-6.1)*	68.8 *(+0.9)*
QKOV-LoRA ($l = 4$)	72.8±0.6 *(+0.9)*	78.0±2.2 *(+11.3)*	64.7±2.1 *(+3.0)*	75.7±0.4 *(-1.7)*	70.8±1.6 *(+0.1)*	68.3±0.7 *(-2.2)*	50.3±3.9 *(-7.7)*	68.7 *(+0.8)*
QKOV-LoRA ($l = 5$)	69.3±0.3 *(-0.8)*	70.8±0.8 *(+4.1)*	60.0±3.6 *(-1.7)*	75.4±0.2 *(-2.0)*	69.2±1.4 *(-1.5)*	66.7±1.0 *(-3.8)*	54.5±1.8 *(-3.5)*	66.6 *(-1.3)*
QKOV-LoRA ($l = 6$)	73.2±0.7 *(+3.1)*	82.7±6.7 *(+16.0)*	63.7±3.7 *(+2.0)*	75.3±0.3 *(-2.1)*	71.6±1.3 *(+1.0)*	68.1±1.7 *(-2.4)*	47.1±0.8 *(-10.9)*	68.8 *(+1.0)*
QKOV-LoRA ($l = 7$)	72.9±0.8 *(+2.8)*	76.8±2.9 *(+10.1)*	64.0±2.9 *(+2.3)*	75.0±0.3 *(-2.4)*	72.3±2.1 *(+1.6)*	67.2±0.3 *(-3.3)*	55.4±3.3 *(-2.6)*	69.1 *(+1.2)*
QKOV-LoRA ($l = 8$)	72.6±0.1 *(+2.5)*	81.0±2.2 *(+14.3)*	63.0±1.6 *(+1.3)*	74.3±0.5 *(-3.1)*	72.0±1.5 *(+1.3)*	66.1±1.1 *(-4.4)*	53.8±2.8 *(-4.2)*	69.0 *(+1.1)*
QKOV-LoRA ($l = 9$)	72.1±0.3 *(+2.1)*	81.5±6.1 *(+14.9)*	62.3±0.9 *(+0.7)*	73.5±0.3 *(-3.9)*	69.7±1.5 *(-1.0)*	67.4±1.6 *(-3.1)*	51.0±3.4 *(-7.0)*	68.2 *(+0.4)*
QKOV-LoRA ($l = 10$)	72.0±0.4 *(+2.0)*	81.5±7.2 *(+14.9)*	62.7±2.6 *(+1.0)*	71.8±0.6 *(-5.6)*	70.9±1.2 *(+0.2)*	66.6±0.4 *(-4.0)*	52.2±1.2 *(-5.8)*	68.2 *(+0.4)*
QKOV-LoRA ($l = 11$)	70.3±0.7 *(+0.2)*	76.8±0.0 *(+10.1)*	62.7±2.1 *(+1.0)*	71.1±0.1 *(-6.3)*	67.5±1.9 *(-3.1)*	64.3±1.7 *(-6.2)*	59.0±3.7 *(+1.0)*	67.4 *(-0.5)*
QK/OV-circuit ($l = 0$)	69.3±0.7 *(-0.7)*	72.0±2.2 *(+5.4)*	58.3±1.2 *(-3.3)*	76.0±0.8 *(-1.4)*	66.5±2.7 *(-4.1)*	68.4±1.3 *(-2.1)*	51.6±4.3 *(-6.4)*	66.0 *(-1.8)*
QK/OV-circuit ($l = 1$)	69.8±0.5 *(-0.3)*	73.8±2.2 *(+7.1)*	58.0±2.2 *(-3.7)*	76.0±0.2 *(-1.4)*	66.3±2.1 *(-4.3)*	68.3±0.6 *(-2.2)*	50.6±1.6 *(-7.4)*	66.1 *(-1.7)*
QK/OV-circuit ($l = 2$)	70.2±1.0 *(+0.1)*	78.0±3.0 *(+11.3)*	62.0±2.2 *(+0.3)*	76.2±0.7 *(-1.2)*	65.9±2.7 *(-4.8)*	67.8±0.8 *(-2.7)*	51.3±1.6 *(-6.7)*	67.4 *(-0.5)*
QK/OV-circuit ($l = 3$)	69.9±0.7 *(-0.2)*	72.6±3.0 *(+5.9)*	61.0±2.8 *(-0.7)*	76.0±0.4 *(-1.4)*	66.8±2.6 *(-3.8)*	68.2±1.0 *(-2.3)*	52.2±0.5 *(-5.8)*	66.7 *(-1.2)*
QK/OV-circuit ($l = 4$)	71.0±0.1 *(+0.9)*	72.0±1.7 *(+5.3)*	61.7±2.6 *(0.0)*	76.4±0.5 *(-1.0)*	68.4±2.1 *(-2.2)*	65.8±1.1 *(-4.8)*	51.6±3.7 *(-6.4)*	66.7 *(-1.2)*
QK/OV-circuit ($l = 5$)	70.5±0.3 *(+0.4)*	76.2±2.2 *(+9.5)*	65.7±2.6 *(+4.0)*	75.7±0.2 *(-1.7)*	68.6±1.6 *(-2.0)*	68.9±0.9 *(-1.6)*	57.1±3.3 *(-1.0)*	68.9 *(+1.1)*
QK/OV-circuit ($l = 6$)	70.4±0.1 *(+0.3)*	69.0±2.2 *(+2.3)*	63.7±1.7 *(+2.0)*	75.3±0.2 *(-2.1)*	69.2±2.5 *(-1.5)*	69.3±0.6 *(-1.2)*	54.2±4.3 *(-3.8)*	67.3 *(-0.6)*
QK/OV-circuit ($l = 7$)	70.4±0.4 *(+0.3)*	72.0±3.4 *(+5.3)*	64.3±1.7 *(+2.6)*	74.4±0.8 *(-3.0)*	71.8±1.8 *(+1.1)*	69.2±0.7 *(-1.3)*	58.7±2.1 *(+0.6)*	68.7 *(+0.8)*
QK/OV-circuit ($l = 8$)	70.2±0.7 *(+0.1)*	73.8±3.7 *(+7.1)*	65.7±3.3 *(+4.0)*	73.3±0.3 *(-4.1)*	72.8±2.5 *(+2.2)*	67.7±1.5 *(-2.9)*	57.1±1.8 *(-1.0)*	68.6 *(+0.8)*
QK/OV-circuit ($l = 9$)	69.4±0.2 *(-0.7)*	73.2±1.5 *(+6.5)*	65.7±3.3 *(+4.0)*	71.6±0.2 *(-5.8)*	69.7±1.4 *(-1.0)*	67.0±1.3 *(-3.5)*	58.0±2.4 *(0.0)*	67.8 *(-0.1)*
QK/OV-circuit ($l = 10$)	66.9±0.1 *(-3.2)*	68.5±0.8 *(+1.9)*	61.0±2.8 *(-0.7)*	70.0±0.3 *(-7.4)*	67.5±1.1 *(-3.1)*	65.4±0.6 *(-5.1)*	59.3±2.5 *(+1.3)*	65.5 *(-2.3)*
QK/OV-circuit ($l = 11$)	65.0±0.5 *(-5.0)*	65.5±2.2 *(-1.2)*	55.3±1.2 *(-6.3)*	68.3±0.0 *(-9.1)*	66.3±0.3 *(-4.3)*	63.4±0.7 *(-7.1)*	60.6±0.0 *(+2.6)*	63.5 *(-4.4)*

- **MultiRC:** 16 epochs, batch size 8, max sequence length 512.
- **RTE:** 40 epochs, batch size 12, max sequence length 256.
- **WSC:** 80 epochs, batch size 8, max sequence length 384.

Optimization and Reproducibility. All models were optimized using the AdamW optimizer (Loshchilov & Hutter, 2017) with a learning rate of 2×10^{-4} and a weight decay of 0.0. A linear learning rate scheduler with a warmup phase corresponding to 10% of the total training steps was employed. To prevent exploding gradients, we clipped the maximum gradient norm at 1.0. To ensure the statistical reliability of our results, all reported scores on the SuperGLUE development set are the mean and standard deviation computed over 10 independent runs with different random seeds.

Detailed SuperGLUE Results. Table 3 provides a comprehensive breakdown of our results on the SuperGLUE development set. For context, we include two key benchmarks: the performance of a fully fine-tuned ALBERT-base-v2, which serves as a practical upper bound, and the results from our strong LoRA baseline ($r = 16$). The table presents an exhaustive evaluation of our proposed QKOV-LoRA and QK/OV-circuit methods (both with $r = 4$), detailing their performance as the perturbation start index, l, is varied from 0 (perturbing all layers) to 11 (perturbing only the final layer).

The results reveal a complex relationship between the start index l and overall performance, highlighting that there is no single optimal value for l across all tasks. The best configuration is highly task-dependent. For instance, while both methods consistently improve performance on CB and BoolQ across a wide range of l values, tasks like WSC and MultiRC often show a performance decrease compared to the LoRA baseline. This suggests that while our approach is beneficial for tasks requiring iterative refinement, a uniform perturbation schedule may be suboptimal for tasks that require highly specialized computations. Comparing the two methods, QKOV-LoRA generally achieves higher peak performance, with its best configuration ($l = 7$) reaching a 69.1 average score. However, the more constrained QK/OV-circuit also demonstrates its effectiveness, surpassing the LoRA baseline at several start indices (e.g., $l = 5, 7, 8$). This detailed analysis supports the main paper's conclusion that strategically introducing depth-wise variation is a potent and parameter-efficient technique, while also underscoring the nuanced trade-offs involved in selecting the optimal perturbation strategy.

A.2 ImageNet Experiments

Base Model Specifications. The base model for all vision experiments was the standard Vision Transformer (ViT-B/16) pretrained on ImageNet-21k and fine-tuned on ImageNet-1k, as described by Dosovitskiy et al. (2020). This model has 12 layers ($L = 12$), a hidden dimension of 768, 12 attention heads, and processes images as a sequence of 16x16 patches.

Dataset and Preprocessing. We conducted large-scale experiments on the ImageNet-1k dataset (Deng et al., 2009). For the low-data regime experiments, models were trained on a randomly selected 10% subset of the full ImageNet-1k training data. All models were evaluated on the standard, full ImageNet-1k validation set. Input images were resized to 224x224 pixels. Pixel values were normalized using the standard ImageNet mean and standard deviation.

Training and Optimization. All models were trained for 100 epochs using a total batch size of 1024 (distributed across GPUs). For both QKOV-LoRA and QK/OV-circuit, we set the LoRA rank to $r = 16$, with a scaling factor of $\alpha = 32$ and a dropout rate of 0.1, as this larger rank was found to be more effective for complex vision tasks. The models were trained using the AdamW optimizer with a base learning rate of 1×10^{-3}, which was scaled linearly with the batch size. We used a weight decay of 0.05 and a cosine learning rate scheduler with a 5-epoch linear warmup. The loss function was the standard Cross-Entropy Loss with label smoothing (smoothing factor of 0.1). To account for training variability, all reported ImageNet accuracies are the mean and standard deviation calculated over 20 independent runs with different random seeds.

A.3 CIFAR-10 Downstream Task Experiments

Fine-tuning Setup. To evaluate transfer learning performance, the ImageNet-trained models (with their respective compressed structures and learned perturbations) were fine-tuned on the CIFAR-10 dataset (Krizhevsky et al., 2009). This process assesses the generalizability of the learned representations.

Data Preprocessing and Augmentation. Input images from CIFAR-10 were resized to 224x224 to match the ViT's expected input resolution. During training, we applied standard data augmentations for CIFAR-10, including random cropping (with padding) and random horizontal flipping. During evaluation, only a center crop was used. Images were normalized using the CIFAR-10 dataset's mean and standard deviation.

Optimization. The models were fine-tuned for 100 epochs with a batch size of 64. We used the AdamW optimizer with a learning rate of 1×10^{-4} and a weight decay of 1×10^{-4}. Gradients were clipped at a maximum norm of 1.0. The learning rate was managed by a cosine annealing scheduler with a 500-step linear warmup phase to ensure stable fine-tuning. The loss function was Cross-Entropy Loss. The base ViT-B/16 model, when fully fine-tuned under this setup, achieves a baseline accuracy of 99.08% on the CIFAR-10 test set.