Mixture-of-Control: State-Aware Fine-Tuning for Transformer-based Models

Duc Anh Nguyen [1*] **Tien Ngoc Luu** [1*] **Tung Pham** [1] **Toan Tran** [1]

Abstract

State-based fine-tuning has emerged as a compelling alternative to weight-based adaptation for transformers, updating lightweight controls into states rather than model weights, offering substantial memory savings while retaining parameter efficiency. However, most existing state-based methods typically apply only per-block control updates, which limits inter-block information exchange and restricts representational adaptation. Meanwhile, prior mechanisms that enable cross-block communication often introduce considerable computational overhead, reducing their practicality for efficient fine-tuning. We introduce Mixture-of-Control (MoC), a lightweight fine-tuning framework that adaptively integrates local and global control signals to enhance representation learning. MoC treats block-wise control states as experts in a sparse mixture-of-experts process, enabling efficient communication across transformer blocks. Empirical results across diverse transformer-based benchmarks demonstrate that MoC outperforms state-based methods while maintaining a comparable memory and computational efficiency.

1. Introduction

Fine-tuning large pretrained models is a standard approach to adapt foundation models to downstream tasks, but full fine-tuning is expensive: it updates and stores billions of parameters per task. Parameter-efficient fine-tuning (PEFT) reduces this cost by freezing the backbone and learning only a small set of task parameters. Among PEFT methods, LoRA (Hu et al., 2022) is particularly effective, replacing weight updates with a low-rank term that largely preserves performance while sharply reducing trainable parameters.

Many LoRA variants–including VeRA (Kopiczko et al., 2023), BinaryLoRA (Zhang et al., 2024a), MoLE (Wu et al., 2024), and DoRA (Liu et al., 2024a)–push parameter or compute efficiency via vectorized updates, binary adaptation, mixture of lora experts, or decomposed updates. However, most remain layer-local, offering limited cross-layer communication and constraining structured representation adaptation. Recently, MoLEx (Teo & Nguyen, 2025) partially addresses this with a Sparse Mixture-of-Layer-Experts, but its Top-K selection requires extra layer-forward passes, raising per-layer cost, especially in large models.

Moreover, existing approaches often reduce trainable parameters without adequately addressing training-time memory usage For example, halving DoRA's rank reduces GPU memory by only 0.256GB on an 8B model (Zhang et al., 2025), showing that fewer trainable parameters need not yield meaningful memory savings. Similarly, existing cross-layer interaction mechanisms such as MoLEx (Teo & Nguyen, 2025) often add substantial computation and memory overhead, limiting their scalability as architectural complexity and model depth increase.

These two issues–restricted cross-layer communication and high memory overhead–motivate a different perspective: fine-tuning via control theory (Zhang et al., 2024c). Rather than modifying system parameters, control systems steer states via feedback. Parallel Control (Zhang et al., 2025) casts LoRA as a control term and activations as states, reducing memory by avoiding intermediate-state storage. Yet it remains local to each layer and lacks global coordination across depth. This motivates our central research question:

Research Question

"How can state-based fine-tuning be designed to capture deep cross-layer interactions without sacrificing its core advantage: memory efficiency?"

We answer this with **Mixture-of-Control (MoC)**, a lightweight state-based method that couples local and global signals via a recurrent mixture-of-experts control pathway. MoC treats each layer's low-rank control module as an *expert* and uses a shared gate to sparsely route each layer's

[1]Qualcomm AI Research. Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.
[*]Equal contribution Correspondence to: Duc Anh Nguyen <nguyenducanh909.bkhn@gmail.com>, Toan Tran <toatran@qti.qualcomm.com>.

Published at ICML 2026 Workshop on Connecting Low-rank Representations in AI, CoLoRAI, Seoul, South Korea. Copyright 2026 by the author(s).

representation to a Top-K set of experts, producing a global control signal. This global signal is then mixed with the layer-local control (e.g., α vs. $1-\alpha$) and injected into the frozen block, enabling adaptive information propagation across depth. Our MoC supports cross-layer communication via low-rank projections without extra pretrained-block passes, preserving state-memory efficiency with minimal overhead. It is architecture-agnostic (FFN or attention; shared or separate gates) and improves robustness and adaptation across benchmarks while matching PEFT-level memory use.

In summary, our contributions are fourfold:

- We identify key limits of existing PEFT and state-based adaptation: controllers are mostly block-local (weak global coordination), while cross-block mechanisms often add substantial, block-dependent compute that impedes scaling to deep/complex architectures.
- We show that cross-block communication can be realized via a lightweight low-rank *control* pathway that exchanges information across layers without reusing pretrained blocks, offering an efficient alternative to full-block interaction.
- We propose **Mixture-of-Control (MoC)**, a stable, memory-efficient state-based framework that integrates block-local control with a sparsely routed global mixture of control experts, supported by theory showing the expressiveness preservation while improving stability, gradient flow, and representation error bound under mild assumptions.
- We conduct extensive experiments across NLU and NLG benchmarks on eight pretrained Transformer backbones, showing that MoC consistently improves adaptation while matching strong PEFT baselines in parameter and memory efficiency.

2. Related Works

Parameter-Efficient Fine-Tuning The rapid growth of Transformer-based models (Vaswani et al., 2017; Dosovitskiy et al., 2020) has made full fine-tuning computationally expensive, motivating Parameter-Efficient Fine-Tuning (PEFT). Early methods focused on tuning selected layers (Oquab et al., 2014; Alain & Bengio, 2018; Lee et al., 2023; Kaplun et al., 2023), while recent approaches introduce lightweight trainable components into pretrained models. Adapter-based methods (Houlsby et al., 2019; Mahabadi et al., 2021) insert bottleneck modules, whereas prompt-based methods (Lester et al., 2021; Liu et al., 2022; Li & Liang, 2021) optimize soft prompts but often suffer from initialization sensitivity (Razdaibiedina et al., 2023).

Low-Rank Adaptation Low-Rank Adaptation methods, most notably LoRA (Hu et al., 2021), approximate weight updates using low-rank matrices to reduce trainable parameters. Subsequent variants improve efficiency via parameter sharing, quantization, or adaptive selection (Kopiczko et al., 2023; Dettmers et al., 2023; Hyeon-Woo et al., 2023; Zhang et al., 2024b). A notable extension involves leveraging a mixture of LoRA experts (Wu et al., 2024), where each layer learns a set of distinct low-rank experts for adaptive representation. More recently, DoRA (Liu et al., 2024a) decomposes weight updates into magnitude and direction components, offering improved performance. Despite these advances, most approaches remain localized to individual layers, providing limited cross-layer communication and constraining the necessary representation during adaptation.

Cross-Layer Communication Adaptation Unlike local adaptation strategies, Mixture of Layer Experts (MoLEx) (Teo & Nguyen, 2025) conceptualizes each pretrained layer (augmented with LoRA) as an expert and performs sparse upcycling through Top-K routing across depth, enabling global information exchange among layers. By conditionally aggregating these layer experts, MoLEx enriches input representations with cross-layer signals, improving adaptability beyond localized updates. However, its Top-K routing mechanism necessitates additional forward passes through the selected pretrained blocks, resulting in significantly higher computational cost than purely local approaches, especially for more complex model architectures.

Control-Theoretic View and State-Based Tuning Recent work has connected PEFT with Control Theory (Kirk, 2004), interpreting deep networks as dynamical systems where layers evolve feature states over time (Li et al., 2018). This perspective has informed research on continuous-depth models, stability, and optimization (Tabuada & Gharesifard, 2024; Lu et al., 2020; Nguyen et al., 2024; Benning et al., 2019). Zhang et al. (Zhang et al., 2024b) formalized LoRA as a controlled dynamical system, with pretrained models defining nominal dynamics and learnable controllers enabling adaptation. Building on this, the State-Based Framework (Zhang et al., 2025) introduces Parallel Control via LoRA, reducing memory usage by avoiding intermediate state storage. Our work generalizes the framework, providing a more scalable state-based adaptation approach with improved performance and comparable memory efficiency.

3. Preliminaries

This section introduces the foundational concepts of the Weight-based method, the State-based method, and the Sparse Mixture of Experts (SMoE).

3.1. Weight-based Fine-Tuning

LoRA (Hu et al., 2022) fine-tunes $W_0 \in \mathbb{R}^{d\times k}$ by adding a low-rank update $\Delta W = BA$ with $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times k}$, and $r \ll \min\{d, k\}$, yielding $W' = W_0 + BA$ where W_0 is frozen and A, B are trainable. Many variants exploit this structure for efficiency, including VeRA (Kopiczko et al., 2023), MoLE (Wu et al., 2024), and DoRA (Liu et al., 2024a), which leverage low-rank structures to reduce computational cost. These mechanisms are categorized as weight-based fine-tuning (Weight-FT), which adapts model weights directly. Its effectiveness is often attributed to the low intrinsic-rank structure of task-specific updates (Hu et al., 2022; Aghajanyan et al., 2021). Mathematically, let x_0 be the input, x_t be the output representation at block t, and a_t be a nonlinear activation (ReLU or GeLU), a pre-trained transformer block can be adapted as follows (Chen et al., 2022):

$$x_{t+1} = x_t + a_t((W_t + \Delta W_t)x_t)\,. \tag{1}$$

Weight-FT often incurs high memory costs due to storing intermediate states. To mitigate this, the State-Based Framework (Zhang et al., 2025) reformulates fine-tuning from a control-theoretic perspective.

3.2. State-based Fine-Tuning

Weight-FT often incurs a high memory cost due to the storage of intermediate states. To mitigate this challenge, the State-Based Framework (Zhang et al., 2024c) reformulates fine-tuning from a control-theoretic perspective.

Forward pass as the feedback control system Consider the continuous-time ordinary differential equations (ODEs) with affine control (Franklin et al., 2002):

$$\dot{x}(t) = f_t(W(t)x(t)) + G(t)u(t), \tag{2}$$

where $x(t)$ is the state vector, $W(t)$ is the state matrix, f_t is the intrinsic dynamic, $u(t)$ is the control input and $G(t)$ is the control matrix. In state-feedback control systems, $u(t)$ can be defined as a function of state $x(t)$ through the feedback gain matrix $K(t)$, such that $u(t) = K(t)x(t)$. Substituting this into the system in Equation (2) yields:

$$\dot{x}(t) = f_t(W(t)x(t)) + G(t)K(t)x(t), \tag{3}$$

where $G(t)K(t)x(t)$ serves as the control term, modifying the system's trajectory in continuous space through the control matrix $G(t)$ and the feedback gain $K(t)$. Discretizing the Equation (3) gives

$$x_{t+1} = x_t + f_t(W_t x_t) + G_t K_t x_t. \tag{4}$$

Considering x_t as the output representation in the t-th block, the feedback control loop can be viewed as analogous to the forward pass in a neural network.

State-based fine-tuning Motivated by this analogy, Zhang et al. (2025) introduced Parallel Control, interpreting $G_t K_t$ as a low-rank control term:

$$x_{t+1} = x_t + f_t(W_t x_t) + g_t(x_t), \tag{5}$$

where g_t is the control function, implemented in practice as a low-rank matrix transformation, i.e., $g_t(x_t) = \Delta W_t x_t$. This design aims to reduce memory usage in intermediate layers while mitigating performance degradation.

3.3. Sparse Mixture of Experts (SMoE)

A SMoE model is typically composed of multiple MoE blocks, each containing a set of experts. The experts within each block are responsible for processing different aspects of the input, and their outputs are combined to form the block's final output. Let $\mathbf{X} \in \mathbb{R}^{m\times d}$ be the matrix of m input token embeddings, where each token is represented by a d-dimensional vector and $\mathbf{W}_{\text{gate}} \in \mathbb{R}^{n\times d}$ be the gating weight matrix for n experts. The gating score matrix $\mathbf{S} \in \mathbb{R}^{m\times n}$ is computed as: $\mathbf{S} = \mathbf{X}\mathbf{W}_{\text{gate}}^{\top}$.

Each entry $s_{i,j}$ in $\mathbf{S}$ represents the compatibility score between token i and expert j, showing how suitable expert j is for processing token i. Higher scores indicate a stronger preference for routing a token to an expert. For each token $i \in \{1, \dots, m\}$, we select the top-k experts with the highest scores, denoted by the index set $\mathcal{T}_i \subseteq \{1, \dots, n\}$ with $|\mathcal{T}_i| = k$. The routing weight for token i to expert j is determined as $w_{i,j} = \begin{cases} \dfrac{\exp(s_{i,j})}{\sum_{j'\in\mathcal{T}_i}\exp(s_{i,j'})} & \text{, if } j \in \mathcal{T}_i, \\ 0 & \text{, otherwise.} \end{cases}$ Each expert j is presented by a feedforward network $f_j : \mathbb{R}^d \to \mathbb{R}^d$. The output of the token i is aggregated as:

$$\mathbf{y}_i = \sum_{j\in\mathcal{T}_i} w_{i,j} \cdot f_j(\mathbf{x}_i). \tag{6}$$

To mitigate routing collapse—where a small subset of experts receives most tokens while the remaining experts are underutilized—several mechanisms have been proposed (Fedus et al., 2022; Zoph et al., 2022; Nguyen et al., 2025). The most widely adopted approach is to incorporate an auxiliary load-balancing loss into the training objective (Fedus et al., 2022; Lepikhin et al., 2020). Specifically, given a sequence of length T, the auxiliary loss is defined as follows:

$$\mathcal{L}_{\text{Balance}} = \alpha \sum_{j=1}^{N} f_j P_j,$$

where $P_j = \frac{1}{T}\sum_{i=1}^{T} \text{softmax}(s_{i:})_j$ and

$$f_j = \frac{n}{km}\sum_{i=1}^{m} \mathbf{1}(\text{token } i \text{ selects expert } j).$$

4. Methodology

This section introduces Mixture-of-Control (MoC), a generalized state-based fine-tuning framework for efficient cross-block communication. Unlike methods requiring $> 2\times$ forward passes through pretrained blocks, MoC uses lightweight control states to enable information transfer with negligible compute overhead, making it practical for large models and complex architectures.

4.1. Generalized State-Based Tuning: Retaining MoLEx Representational Power

We begin from the following generalized state-based update:

$$x_{t+1} = x_t + f_t(W_t x_t) + \alpha\, g_t(x_t) + (1-\alpha)\, g_{h_t}(x_t), \tag{7}$$

where W_t denotes the pretrained weight of block t, g_t is its local low-rank control, g_{h_t} is a control chosen from another block h_t, and $\alpha \in [0,1]$ balances local and cross-layer information. Standard state-based tuning is recovered when $h_t = t$, while $h_t \neq t$ enables cross-block communication through lightweight low-rank controls rather than full-block reuse.

The adoption of low-rank controls is motivated by (Zhang et al., 2025), which shows that they efficiently adapt block representations as an alternative to Weight-FT. In constrast to MoLEx (Teo & Nguyen, 2025), which relies on pretrained block reuse under Weight-FT, our method facilitates lightweight cross-block communication while substantially reducing training time and memory overhead. This design choice is further supported by (Zeng & Lee, 2023), which demonstrates that sufficiently expressive LoRA adapters can approximate arbitrary target functions.

We analyze linear and nonlinear smooth f_t. Generalized state-based fine-tuning retains MoLEx expressivity: it is exact for linear f_t (under mild conditions) and bounded-approximate for nonlinear f_t. We present the linear proof for $f_t(x) = x$; general linear maps follow directly.

Theorem 4.1 (MoLEx Expressivity via State Controls)**.** *Consider the MoLEx update over L blocks defined by*

$$\begin{aligned} z_{t+1} &= z_t + \alpha f_t((W_t + \Delta W_t')z_t) \\ &\quad + (1-\alpha) f_{h_t}\big((W_{h_t} + \Delta W_{h_t}')z_t\big), \end{aligned} \tag{8}$$

where z_t is the output of the t-th block and the generalized state-based update defined in Equation (7). Assuming that $\alpha > \frac{1}{2}$ and $z_1 = x_1$, then there exist controls $g_k : \mathbb{R}^d \to \mathbb{R}^d$ such that: when f_k is linear we have $z_t = x_t$ for all $t \in \{1, \ldots, L+1\}$; moreover, when f_k is smooth nonlinear under small perturbations, i.e., $\forall t$, $\|\Delta W_t'\| < \xi$ ($\xi > 0$ and sufficiently small), then $\mathbb{E}\big[\|z_t - x_t\|^2\big]$ is bounded and vanishes as $\xi \to 0$.

This theorem indicates that, under mild assumptions, generalized state-based updates can preserve MoLEx-induced representations through state controls. Motivated by this expressivity guarantee, we next develop an adaptive state-based framework that enables inter-block communication.

4.2. Mixture-of-Control (MoC) for State-based Fine-Tuning

As discussed in Section 4.1, inter-block communication can be implemented via low-rank control modules instead of reusable pretrained blocks, substantially reducing computation. We propose Mixture-of-Control (MoC), a lightweight, SMoE-inspired fine-tuning method that captures global information in a depth-adaptive manner. MoC treats the control modules $\{g_k\}_{k=1}^{L}$ as experts and employs a shared gating matrix $\mathbf{W}_{\text{gate}}$ across layers. At each block, the input selects Top-K experts to produce a global control signal; this signal is then combined with local control and added to the pretrained block output before passing to the subsequent block (see Figure 1).

Formally, MoC coordinates local and global control signals adaptively for efficient inter-block information exchange, instantiating Equation (7) as:

$$x_{t+1} = x_t + f_t(W_t x_t) + \alpha g_t(x_t) + (1-\alpha) \sum_{k=1}^{L} \pi_k(x_t) g_k(x_t), \tag{9}$$

where

$$\pi_k(x_t) = \mathrm{softmax}(\mathrm{TopK}(\mathbf{W}_{\text{gate}} x_t))_k \tag{10}$$

denotes the routing weight for control expert k, L is the number of control sites, and $\mathbf{W}_{\text{gate}}$ is the shared routing matrix. The coefficient α balances local token-specific information with global information injected by routed control experts. In our experiments, we use top-K routing with $K = 1$, with ablations on K in Section 5.3. To improve training efficiency for Transformer models, we employ *batch-level routing*, in which token-level routing scores are aggregated using either the mean or the mode. During autoregressive inference, each sample is routed independently based solely on its available prefix. Additional routing details are provided in Appendix B of Appendix.

MoC extends state-based fine-tuning by combining block-local controls with routed control experts for lightweight inter-block exchange. For Transformers, we study four variants: **MoC (FFN)** routes controls in FFN blocks, and **MoC (ATTN)** in attention blocks. For finer control, **MoC (ATTN+FFN, w/o sharing)** uses separate gates for attention and FFN, while **MoC (ATTN+FFN)** shares a single gate to coordinate both. Unless noted otherwise, **MoC** refers to **MoC (ATTN+FFN)**.

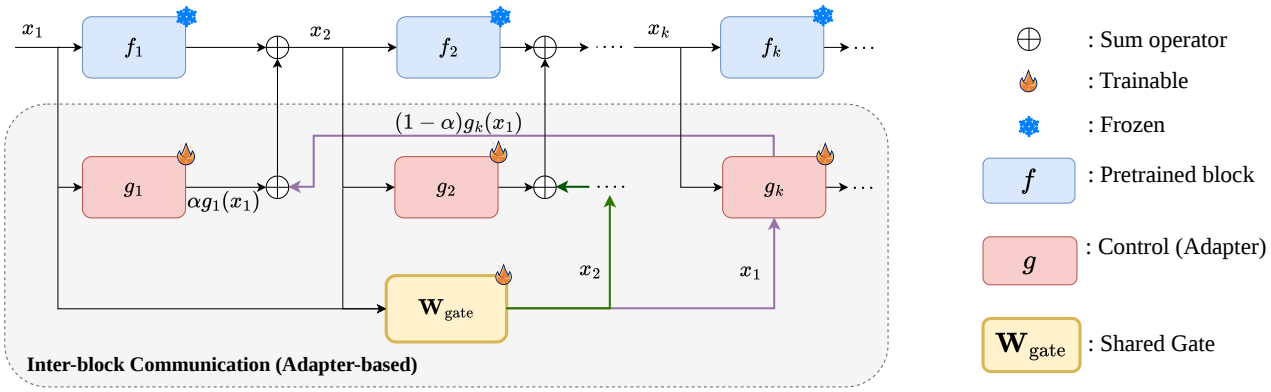


Figure 1. Overview of our Mixture-of-Control (MoC): For an arbitrary input x at the block i, the pretrained block f_i and the local control block g_i process the representation in parallel. A shared gate selects the Top-K control block from all layers, and the selected controller is fused with the local one to form the block control signal. The figure illustrates this mechanism using x_1 and later x_2 as the input under Top-K=1.

Sharing a gate across attention and FFN follows fine-grained SMoE: granular routing improves expert specialization and utilization without additional cost (Krajewski et al., 2024; Lo et al., 2025). In MoC, a shared gate jointly coordinates both pathways for finer adaptation at similar overhead. Overall, MoC enables depth-adaptive Top-K information exchange, increases capacity under a frozen backbone, and remains a lightweight, architecture-agnostic drop-in for standard fine-tuning.

We rigorously evaluate MoC through a theoretical analysis of *stability*, *gradient flow*, and *representation error*, and compare the results to standard Parallel Control.

4.3. Theoretical Analysis

Stability Analysis. We first examine forward stability, since controlling the growth of hidden states is a prerequisite for stable optimization.

Proposition 4.2 (Stability with Top-K Routing)**.** *Consider a network with L control sites, where each control is a low-rank projection $\Delta W_k = B_k A_k$ with $\|B_k\| \le \beta_k$ and $\|A_k\| \le \gamma$. Let the router select a top-K index set $S_K(x_t)$ with weights $\pi_k(x_t)$ given in Equation (10). Define the effective routed magnitude as $\beta_K(x_t) = \sum_{k \in S_K(x_t)} \pi_k(x_t)\beta_k$.*

Let $\{x_t^c\}_{t=1}^{L+1}$ and $\{x_t\}_{t=1}^{L+1}$ denote the representations produced by Parallel Control and MoC, respectively. Then for each t, $\|x_{t+1}^c\| \le (1 + L_f + \beta_t\gamma)\|x_t^c\|$, whereas MoC with mixing coefficient $\alpha \in [0,1]$ satisfies $\|x_{t+1}\| \le \big(1 + L_f + \alpha\beta_t\gamma + (1-\alpha)\beta_K(x_t)\gamma\big)\|x_t\|$, where L_f bounds the Lipschitz contribution of the frozen dynamics. Moreover, if $x_1 = x_1^c$ and the router selects experts such that $\mathbb{E}[\beta_K(x_t)|x_t] < \beta_t$, then MoC admits a strictly tighter expected one-step growth factor:

$$1 + L_f + \big[\alpha\beta_t + (1-\alpha)\mathbb{E}[\beta_K(x_t)|x_t]\big]\gamma < 1 + L_f + \beta_t\gamma.$$

*Thus, effective routing **yields more stable forward dynamics** by replacing a fixed local control norm with a data-dependent routed norm.*

The key quantity in this result is the routed norm $\beta_K(x_t)$. We enforce $\mathbb{E}[\beta_K(x_t)|x_t] < \beta_t$ by using a load-balancing loss (Fedus et al., 2022; Lepikhin et al., 2020), promoting balanced expert use and better control-expert selection.

Gradient Flow. A tighter bound on the expected forward growth controls the layerwise Jacobian norms in the back-propagated product, thereby improving gradient stability as depth increases.

Corollary 4.3 (Improved Gradient Flow)**.** *For each time step t, denote by $J_t = \frac{\partial x_{t+1}}{\partial x_t}$ and $J_t^c = \frac{\partial x_{t+1}^c}{\partial x_t^c}$ the Jacobians of the forward dynamics for the MoC and Parallel Control path, respectively. Under Proposition 4.2's assumptions, MoC admits a tighter expected Jacobian norm upper bound than its Parallel Control counterpart. In particular, $\forall t$, we have $\|J_t^c\| \le M_t^c$ and $\|J_t\| \le M_t$, where $M_t = 1 + L_f\|W_t\| + \alpha\beta_t\gamma + (1-\alpha)\beta_K(x_t)\gamma$ and $M_t^c = 1 + L_f\|W_t\| + \beta_t\gamma$. Moreover, $M_t \le M_t^c, \forall t$, and consequently,*

$$\prod_{t=1}^{L}\|J_t\| \le \prod_{t=1}^{L} M_t \le \prod_{t=1}^{L} M_t^c, \tag{11}$$

*indicating a **more stable gradient flow** under backpropagation in the MoC path.*

Representation Error. Finally, we link adaptive routing to representation quality: by routing and combining multiple control experts, MoC yields a closer approximation to the ideally adaptive control policy than block-local control alone.

Proposition 4.4 (Representation Error Bound)**.** *Consider a network comprising L blocks. Assume $\Delta W^* x$ is*

the optimal control signal that yields task-optimal representations. Denote selected experts exhibit an average error of $\bar{\epsilon}_K^2 := \mathbb{E}[\|\sum_{k\in S_K(x)} \pi_k(x)\Delta W_k x - \Delta W^* x\|^2|x]$, $\bar{\epsilon}_c^2 := \mathbb{E}\left[\|\Delta W_t x - \Delta W^* x\|^2|x\right]$ *and* $\bar{\epsilon}^2 := \mathbb{E}\left[\|g_{moc}(x) - \Delta W^* x\|^2|x\right]$ *are the representation errors of Parallel Control and MoC, respectively. Then the representation error of MoC can be bounded as:*

$$\bar{\epsilon}^2 \leq \alpha\bar{\epsilon}_c^2 + (1-\alpha)\bar{\epsilon}_K^2. \tag{12}$$

Furthermore, under Proposition 4.2's assumptions, $\bar{\epsilon}_K^2 < \bar{\epsilon}_c^2$, *which leads to* $\bar{\epsilon}^2 < \bar{\epsilon}_c^2$.

These results show a connected benefit of sparse state-control routing: effective routing tightens forward stability bounds, which improves gradient propagation, while the same routed mixture reduces the gap to the ideal representation. The detailed proofs of Theorem 4.1, Proposition 4.2, Corollary 4.3, and Proposition 4.4 are provided in Appendix (see Appendix A).

5. Experiments

This section details the experimental setup, presents the primary results of our proposed method compared with state-of-the-art baselines, and includes an ablation study.

5.1. Settings

Datasets. We conduct experiments on diverse language modeling benchmarks. For natural language generation (NLG), we use 8 commonsense reasoning datasets: BoolQ (Clark et al., 2019), PiQA (Bisk et al., 2020), Social IQa (Sap et al., 2019), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC-E/ARC-C (Clark et al., 2018), and OpenBookQA (Mihaylov et al., 2018). For natural language understanding (NLU), we use GLUE (Wang et al., 2018), which includes datasets such as SST-2, MRPC, CoLA, QNLI, RTE, and STS-B, designed to access the ability to interpret linguistic structure, semantic relationships, and contextual meaning. These benchmarks provide a comprehensive evaluation across language tasks.

Backbone models. For NLG, we use LLaMA2-7B (Touvron et al., 2023), LLaMA3-8B (Grattafiori et al., 2024), Mistral-7B Instruct (Jiang et al., 2023), and Qwen2.5-14B Instruct (Qwen et al., 2025). For NLU, we evaluate RoBERTa and DeBERTa in both base and large variants.

Baselines. Across domains, we compare against representative adaptation methods, including LoRA (Hu et al., 2021), DoRA (Liu et al., 2024b), MoLEx (Teo & Nguyen, 2025), and the recent Parallel Control (Zhang et al., 2025) (abbreviated as Control).

Evaluation Metrics and Analysis. For NLG tasks, we evaluate performance using accuracy for eight commonsense reasoning datasets. For NLU tasks, we follow standard evaluation protocols and report task-specific metrics: Matthew's correlation coefficient for CoLA, Pearson correlation for STS-B, and accuracy for all remaining tasks.

5.2. Main Results

NLG Tasks. We evaluate MoC on NLG tasks (Table 1). MoC consistently outperforms prior methods across backbones from LLaMA2-7B to Qwen2.5-14B, beating LoRA by 2.4%–6.0% and the strongest baseline by 1.0%–3.0% with comparable compute. While MoLEx also enables cross-block communication, it costs up to $> 2\times$ time and memory on Qwen2.5-14B; MoC achieves better performance with superior scalability. Beyond accuracy, Figure 2 reports inference throughput across model scales and batch sizes. While state-based methods remain less efficient than fully merged Weight-FT, MoC consistently outperforms MoLEx in throughput—even after merging—improving the accuracy–efficiency trade-off and narrowing the gap to mergeable fine-tuning.

NLU Tasks. Next, we evaluate MoC against prior approaches on the GLUE benchmark in Table 2. MoC consistently outperforms competing baselines across all six datasets, achieving average gains of 1.5% and 2.0% over LoRA for RoBERTa-Base and RoBERTa-Large, respectively, and surpassing the strongest competing baseline by 1.2%. Similarly, MoC outperforms prior methods by 1.5% and 1.2% on DeBERTa-Base and DeBERTa-Large, respectively. Together, these results demonstrate the effectiveness of MoC across both NLG and NLU settings.

5.3. Ablation Studies

TopK Selection. We analyze Top-K selection across LLM backbones and report training time and GPU memory on Mistralv3-7B to evaluate scalability. As shown in Figure 3(b), MoC performs best with two activated experts, followed by eight. In MoC, Top-$K = 1$ pairs the block-local controller with one routed controller from other blocks, similar to Top-2 routing in sparse MoE and offering a strong accuracy–efficiency trade-off (Lepikhin et al., 2020; Teo & Nguyen, 2024; Zhou et al., 2023). Unlike standard MoE, MoC selects experts across blocks rather than within a layer, so increasing Top-K changes both control capacity and inter-block information exchange. We also test $K \in \{0, 1, 3, 7\}$ (correspond to the number of activated experts $\in \{1, 2, 4, 8\}$), where $K = 0$ uses only the local controller (*Control baseline*); see Figure 3(a). Training time and memory increase moderately with Top-K and remain well below MoLEx in Table 1, indicating efficient scaling. Routing visualizations further show adaptive use of local

Table 1. Comparison of algorithm performance across commonsense reasoning benchmarks. * marks previously published numbers. Best results are in **bold**.

Model	Method	% Params	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg (↑)	Time (↓) (h)	GPU Mem. (↓) (GB)
LLaMA2-7B	LoRA*	0.83	69.8	79.9	79.5	83.6	82.6	79.8	64.7	81.0	77.6	6.1	48.0
	DoRA*	0.85	71.8↑2.0	**83.7**↑3.8	76.0↓3.5	89.1↑5.5	82.6↑0.0	83.7↑3.9	68.2↑3.5	**82.4**↑1.4	79.7↑2.1	10.5	80.2
	MoLEx	0.85	69.7↓0.1	81.9↑2.0	78.4↓1.1	91.3↑7.7	81.5↓1.1	81.5↑1.7	65.8↑1.1	80.8↓0.2	78.9↑1.3	9.8	77.4
	Control*	0.50	**72.3**↑2.5	82.5↑2.6	79.2↓0.3	89.1↑5.5	83.1↑0.5	83.0↑3.2	68.5↑3.8	79.0↓2.0	79.6↑2.0	4.2	42.6
	MoC	0.51	71.8↑2.0	83.2↑3.3	**79.6**↑0.1	**93.3**↑9.7	**84.2**↑1.6	83.6↑3.8	68.6↑3.9	80.4↓0.6	**80.5**↑2.9	5.5	47.6
LLaMA3-8B	LoRA*	0.71	70.8	85.2	79.9	91.7	84.3	84.2	71.2	79.0	80.8	7.5	59.1
	DoRA*	0.71	74.6↑3.8	89.3↑4.1	79.9↑0.0	95.5↑3.8	85.6↑1.3	90.5↑6.3	80.4↑9.2	85.8↑6.8	85.2↑4.4	10.5	69.1
	MoLEx	0.72	70.1↓0.7	86.2↑1.0	80.4↑0.5	93.2↑1.5	84.1↓0.2	84.8↑0.6	71.3↑0.1	82.2↑3.2	81.5↑0.7	11.0	94.9
	Control*	0.42	74.1↑3.3	87.8↑2.6	80.7↑0.8	95.5↑3.8	86.0↑1.7	90.8↑6.6	80.0↑8.8	87.8↑8.8	85.3↑4.5	4.4	54.0
	MoC	0.43	**75.5**↑4.7	**89.5**↑4.3	**82.0**↑2.1	**96.6**↑4.9	**88.8**↑4.5	**91.5**↑7.3	**81.6**↑10.4	**88.0**↑9.0	**86.7**↑5.9	6.5	56.7
Mistralv3-7B	LoRA	0.78	68.7	83.5	77.2	90.3	82.5	85.7	73.4	80.5	80.2	6.5	58.4
	DoRA	0.79	70.3↑1.5	84.5↑1.0	77.8↑0.6	89.9↓0.4	83.1↑0.5	87.1↑1.4	72.2↓1.2	82.4↑1.9	80.9↑0.7	10.1	70.4
	MoLEx	0.80	71.4↑2.7	85.4↑2.0	78.2↑1.0	90.3↓0.0	83.5↑1.0	84.1↓1.6	75.0↑1.5	82.7↑2.2	81.3↑1.1	10.5	82.3
	Control	0.46	70.6↑1.8	86.9↑3.5	79.0↑1.8	94.2↑3.9	84.8↑2.2	87.9↑2.2	76.5↑3.0	84.8↑4.3	83.1↑2.9	4.3	53.2
	MoC	0.48	**74.2**↑5.5	**88.9**↑5.4	**81.0**↑3.8	**95.6**↑5.3	**87.9**↑5.4	**88.9**↑3.2	**79.5**↑6.1	**87.8**↑7.3	**85.5**↑5.3	5.5	55.6
Qwen2.5-14B	LoRA	0.63	73.4	90.4	82.2	96.0	88.6	94.7	86.1	90.6	87.7	9.4	62.1
	DoRA	0.64	75.1↑1.7	90.1↓0.3	82.0↓0.2	96.3↑0.3	89.4↑0.8	95.0↑0.3	88.2↑2.1	93.0↑2.4	88.6↑0.9	14.3	94.7
	MoLEx	0.64	73.5↑0.1	90.5↑0.1	81.9↓0.3	95.5↓0.5	88.2↓0.4	94.4↓0.3	85.8↓0.3	92.2↑1.6	87.8↑0.1	20.3	124.8
	Control	0.43	72.8↓0.6	88.8↓1.6	81.3↓0.9	94.6↓1.4	87.5↓1.1	94.4↓0.3	86.9↑0.8	91.2↑0.6	87.2↓0.5	6.5	58.7
	MoC	0.43	**75.5**↑2.1	**91.8**↑1.4	**82.4**↑0.2	**96.8**↑0.8	**91.0**↑2.4	**96.6**↑1.9	**90.2**↑4.1	**96.4**↑5.8	**90.1**↑2.4	8.0	61.3

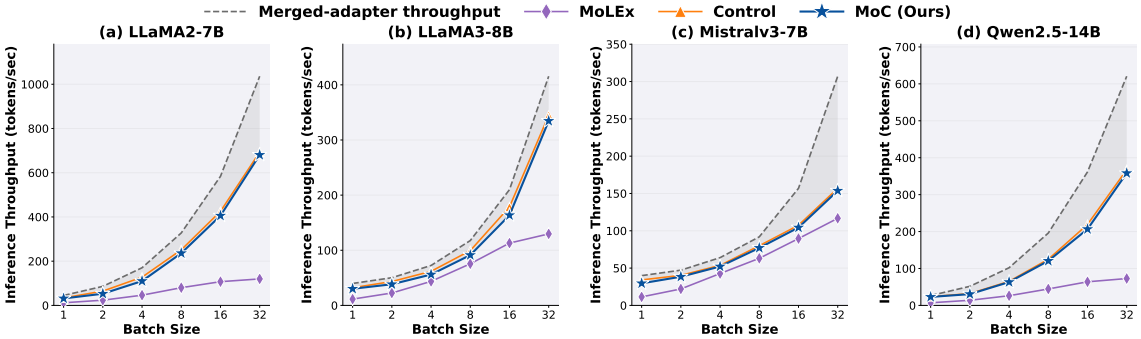


Figure 2. Inference Throughput of MoC compared to prior methods across LLMs and batch size.

and global information, supporting cross-block control.

Mixing Coefficient α. We study the sensitivity of MoC to the mixing coefficient α. Motivated by Theorem 4.1, we vary $\alpha \in \{0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99\}$ while fixing $\lambda = 0.01$. As shown in Figure 3 (c), MoC achieves the best performance at $\alpha = 0.95$ across different LLMs. It also remains competitive or superior to the baselines in Table 1 over a wide range of α values, indicating robustness to α while benefiting from the expressive inter-block communication characterized in Theorem 4.1.

Load-Balancing Coefficient λ. We study the sensitivity of MoC to the load-balancing coefficient λ by fixing $\alpha = 0.95$ and varying $\lambda \in 0, 0.005, 0.01, \ldots, 0.03$, where $\lambda = 0$ disables load balancing. As shown in Figure 3 (d), load balancing improves performance, suggesting better expert utilization and routing, consistent with Proposition 4.2, Corollary 4.3, and Proposition 4.4. Performance peaks at $\lambda = 0.01$, while larger values begin to hinder the main objective.

MoC Variants. We compare MoC variants that control attention, FFN, or both (Table 3); FFN control consistently beats attention-only, and MoC (FFN+ATTN) with a shared gate performs best, matching the view that attention routes information while FFNs drive nonlinear semantic transformation (Geva et al., 2021; Lu et al., 2019). Shared gating over both modules enables richer cross-layer control and finer expert selection, strengthening cross-layer communica-

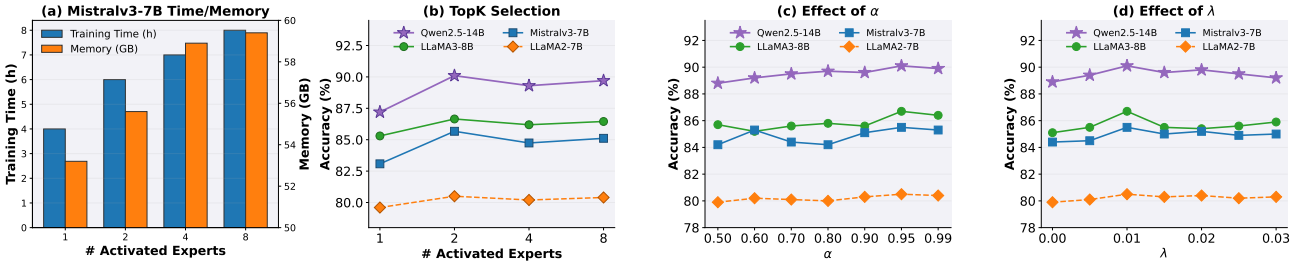


Figure 3. Ablation analysis of MoC under different configurations. (a) Training-time and GPU-memory overhead as the number of activated experts varies, reported on Mistralv3-7B. (b–d) Performance analysis of MoC across different LLM backbones, including (b) TopK Selection, (c) the effect of the coefficient α, and (d) the effect of the coefficient λ.

Table 2. GLUE results across RoBERTa and DeBERTa backbones. Values are medians over five runs; * denotes previously published results. Best results are in **bold**. Superscripts ↑ and ↓ indicate improvement and degradation relative to LoRA.

Model	Method	SST-2	MRPC	CoLA	QNLI	RTE	STS-B	Avg (↑)
RoBERTa-Base	LoRA*	95.1	89.7	63.4	93.3	78.4	91.5	85.2
	DoRA*	95.0↓0.1	89.7↑0.0	64.9↑1.5	92.9↓0.4	79.2↑0.8	91.3↓0.2	85.5↑0.3
	MoLEx*	95.4↑0.3	89.8↑0.1	64.8↑1.4	93.2↓0.1	77.3↓1.1	91.0↓0.5	85.3↑0.1
	Control*	95.3↑0.2	89.3↓0.4	65.3↑1.9	93.1↓0.2	76.9↓1.5	90.6↓0.9	85.1↓0.1
	MoC	**95.6**↑0.5	**90.7**↑1.0	**66.7**↑3.3	**93.4**↑0.1	**81.9**↑3.5	**91.5**↑0.0	**86.7**↑1.5
RoBERTa-Large	LoRA*	96.2	90.2	68.2	94.8	85.2	92.3	87.8
	DoRA*	96.6↑0.4	90.9↑0.7	68.8↑0.6	95.2↑0.4	84.7↓0.5	92.5↑0.2	88.1↑0.3
	MoLEx*	96.4↑0.2	91.4↑1.2	68.2↑0.0	94.8↑0.0	87.1↑1.9	92.0↓0.3	88.3↑0.5
	Control*	96.7↑0.5	90.9↑0.7	69.9↑1.7	95.2↑0.4	86.3↑1.1	92.5↑0.2	88.6↑0.8
	MoC	**97.2**↑1.0	**92.6**↑2.4	**71.4**↑3.2	**95.2**↑0.4	**89.9**↑4.7	**92.5**↑0.2	**89.8**↑2.0
DeBERTa-Base	LoRA*	96.2	89.4	70.4	94.3	83.4	91.0	87.5
	DoRA*	96.0↓0.2	90.2↑0.8	71.2↑0.8	94.3↑0.0	84.0↑0.6	91.0↑0.0	87.8↑0.3
	MoLEx	96.0↓0.2	90.1↑0.7	71.0↑0.6	94.3↑0.0	85.8↑2.4	91.4↑0.4	88.1↑0.6
	Control	96.3↑0.1	89.8↑0.4	71.3↑0.9	94.4↑0.1	84.8↑1.4	91.5↑0.5	88.0↑0.5
	MoC	**96.9**↑0.7	**91.0**↑1.6	**71.8**↑1.4	**94.5**↑0.2	**87.8**↑4.4	**91.9**↑0.9	**89.0**↑1.5
DeBERTa-Large	LoRA*	96.9	91.8	75.0	95.9	91.0	92.7	90.5
	DoRA*	96.6↓0.3	92.1↑0.3	76.9↑1.9	95.8↓0.1	91.1↑0.1	92.6↓0.1	90.9↑0.4
	MoLEx	96.6↓0.3	91.9↑0.1	75.9↑0.9	95.9↑0.0	91.6↑0.6	92.7↑0.0	90.8↑0.3
	Control	**96.7**↓0.2	**92.7**↑0.9	76.4↑1.4	95.9↑0.0	90.0↓1.0	92.3↓0.4	90.6↑0.1
	MoC	**97.3**↑0.4	**92.7**↑0.9	**77.0**↑2.0	**96.2**↑0.3	**93.9**↑2.9	**92.9**↑0.2	**91.7**↑1.2

tion in line with fine-grained SMoE principles (Section 4.2).

Table 3. Comparison of different MoC variants on LLaMA3-8B and Mistralv3-7B.

MoC Variant	LLaMA3-8B (↑)	Mistralv3-7B (↑)
ATTN	85.9	84.9
FFN	86.2	85.1
ATTN+FFN w/o sharing	86.4	84.9
ATTN+FFN	**86.7**	**85.5**

6. Conclusion

In this paper, we identify two limitations of efficient fine-tuning: restricted multi-block coordination in state-based methods and the high overhead of existing cross-block communication approaches. We propose Mixture-of-Control (MoC), a lightweight framework that treats block-wise control states as experts and adaptively combines local and global signals through sparse routing. Across diverse backbones, MoC consistently outperforms prior baselines with minimal training overhead and competitive inference throughput. These results highlight control-driven cross-block communication as a promising direction for scalable Transformer adaptation, while further improving inference efficiency and exploring more expressive control architecture stands for an important direction for future work.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Aghajanyan, A., Gupta, S., and Zettlemoyer, L. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pp. 7319–7328, 2021.

Alain, G. and Bengio, Y. Understanding intermediate layers using linear classifier probes, 2018. URL https://arxiv.org/abs/1610.01644.

Benning, M., Celledoni, E., Ehrhardt, M. J., Owren, B., and Schönlieb, C.-B. Deep learning as optimal control problems: models and numerical methods, 2019. URL https://arxiv.org/abs/1904.05657.

Bisk, Y., Zellers, R., Gao, J., Choi, Y., et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

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

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms, 2023. URL https://arxiv.org/abs/2305.14314.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL https://arxiv.org/abs/2010.11929.

Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

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

Geva, M., Schuster, R., Berant, J., and Levy, O. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, 2021.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

He, P., Liu, X., Gao, J., and Chen, W. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

He, S., Sun, G., Shen, Z., and Li, A. What matters in transformers? not all attention is needed. *arXiv preprint arXiv:2406.15786*, 2024.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp, 2019. URL https://arxiv.org/abs/1902.00751.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Hyeon-Woo, N., Ye-Bin, M., and Oh, T.-H. Fedpara: Low-rank hadamard product for communication-efficient federated learning, 2023. URL https://arxiv.org/abs/2108.06098.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Kaplun, G., Gurevich, A., Swisa, T., David, M., Shalev-Shwartz, S., and Malach, E. Less is more: Selective layer finetuning with subtuning. *arXiv preprint arXiv:2302.06354*, 2023.

Kirk, D. E. *Optimal control theory: an introduction*. Courier Corporation, 2004.

Kopiczko, D. J., Blankevoort, T., and Asano, Y. M. Vera: Vector-based random matrix adaptation. *arXiv preprint arXiv:2310.11454*, 2023.

Krajewski, J., Ludziejewski, J., Adamczewski, K., Pióro, M., Krutul, M., Antoniak, S., Ciebiera, K., Król, K., Odrzygóźdź, T., Sankowski, P., et al. Scaling laws for fine-grained mixture of experts. *arXiv preprint arXiv:2402.07871*, 2024.

Lee, Y., Chen, A. S., Tajwar, F., Kumar, A., Yao, H., Liang, P., and Finn, C. Surgical fine-tuning improves adaptation to distribution shifts, 2023. URL https://arxiv.org/abs/2210.11466.

Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., and Chen, Z. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning, 2021. URL https://arxiv.org/abs/2104.08691.

Li, Q., Chen, L., Tai, C., and E, W. Maximum principle based algorithms for deep learning, 2018. URL https://arxiv.org/abs/1710.09513.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation, 2021. URL https://arxiv.org/abs/2101.00190.

Liu, S.-Y., Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C. F., Cheng, K.-T., and Chen, M.-H. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*, 2024a.

Liu, S.-Y., Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C. F., Cheng, K.-T., and Chen, M.-H. Dora: Weight-decomposed low-rank adaptation, 2024b. URL https://arxiv.org/abs/2402.09353.

Liu, X., Ji, K., Fu, Y., Tam, W., Du, Z., Yang, Z., and Tang, J. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 61–68, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.8. URL https://aclanthology.org/2022.acl-short.8/.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Lo, K. M., Huang, Z., Qiu, Z., Wang, Z., and Fu, J. A closer look into mixture-of-experts in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4427–4447, 2025.

Lu, Y., Li, Z., He, D., Sun, Z., Dong, B., Qin, T., Wang, L., and Liu, T.-Y. Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv preprint arXiv:1906.02762*, 2019.

Lu, Y., Zhong, A., Li, Q., and Dong, B. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations, 2020. URL https://arxiv.org/abs/1710.10121.

Mahabadi, R. K., Henderson, J., and Ruder, S. Compacter: Efficient low-rank hypercomplex adapter layers, 2021. URL https://arxiv.org/abs/2106.04647.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

Nguyen, D. A., Ta, H. B., Duc, N. L., Nguyen, T. M., and Tran, T. Selective sinkhorn routing for improved sparse mixture of experts. *arXiv preprint arXiv:2511.08972*, 2025.

Nguyen, T., Uribe, C. A., Nguyen, T. M., and Baraniuk, R. G. Pidformer: Transformer meets control theory, 2024. URL https://arxiv.org/abs/2402.15989.

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1717–1724, 2014. doi: 10.1109/CVPR.2014.222.

Qwen, :, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., and Qiu, Z. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Razdaibiedina, A., Mao, Y., Hou, R., Khabsa, M., Lewis, M., and Almahairi, A. Progressive prompts: Continual learning for language models, 2023. URL https://arxiv.org/abs/2301.12314.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Sap, M., Rashkin, H., Chen, D., Le Bras, R., and Choi, Y. Social IQa: Commonsense reasoning about social interactions. In Inui, K., Jiang, J., Ng, V., and Wan, X.

(eds.), *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. 4463–4473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1454. URL https://aclanthology.org/D19-1454/.

Tabuada, P. and Gharesifard, B. Universal approximation power of deep residual neural networks via nonlinear control theory, 2024. URL https://arxiv.org/abs/2007.06007.

Teo, R. and Nguyen, T. M. MomentumSMoe: Integrating momentum into sparse mixture of experts. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=y929esCZNJ.

Teo, R. S. and Nguyen, T. M. Molex: Mixture of layer experts for finetuning with sparse upcycling. *arXiv preprint arXiv:2503.11144*, 2025.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pp. 353–355, 2018.

Wu, X., Huang, S., and Wei, F. Mixture of lora experts. *arXiv preprint arXiv:2404.13628*, 2024.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Zeng, Y. and Lee, K. The expressive power of low-rank adaptation. *arXiv preprint arXiv:2310.17513*, 2023.

Zhang, C., Cheng, J., and Li, Q. An optimal control view of lora and binary controller design for vision transformers. In *European Conference on Computer Vision*, pp. 144–160. Springer, 2024a.

Zhang, C., Jingpu, C., Xu, Y., and Li, Q. Parameter-efficient fine-tuning with controls. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 59066–59079. PMLR, 21–27 Jul 2024b. URL https://proceedings.mlr.press/v235/zhang24y.html.

Zhang, C., Jingpu, C., Xu, Y., and Li, Q. Parameter-efficient fine-tuning with controls. In *Forty-first International Conference on Machine Learning*, 2024c.

Zhang, C., Lianhai, R., Cheng, J., and Li, Q. From weight-based to state-based fine-tuning: Further memory reduction on lora with parallel control. In *Forty-second International Conference on Machine Learning*, 2025.

Zhou, Y., Du, N., Huang, Y., Peng, D., Lan, C., Huang, D., Shakeri, S., So, D., Dai, A. M., Lu, Y., et al. Brainformers: Trading simplicity for efficiency. In *International conference on machine learning*, pp. 42531–42542. PMLR, 2023.

Zoph, B., Bello, I., Kumar, S., Du, N., Huang, Y., Dean, J., Shazeer, N., and Fedus, W. St-moe: Designing stable and transferable sparse expert models, 2022. URL https://arxiv.org/abs/2202.08906.

Suplementary Material to "Mixture-of-Control: State-Aware Fine-Tuning for Transformer-based Models"

A. Proofs for Theoretical Results

A.1. Theorem 4.1

Proof. For simplicity, in the linear case, we take f_t to be an identity-like mapping. Thus, MoLEx update is simplified into:

$$z_{t+1} = z_t + [\alpha(W_t + \Delta W'_t) + (1-\alpha)(W_{h_t} + \Delta W'_{h_t})]z_t \tag{13}$$
$$= z_t + W_t z_t + [\alpha(\Delta W'_t) + (1-\alpha)(W_{h_t} - W_t + \Delta W'_{h_t})]z_t \tag{14}$$

Denote $M_t = \alpha(\Delta W'_t) + (1-\alpha)(W_{h_t} - W_t + \Delta W'_{h_t})$, for $t = \overline{1, L}$. Similarly, the generalized state-based update is formulated as follows:

$$\begin{aligned} x_{t+1} &= x_t + W_t x_t + \alpha \Delta W_t x_t + (1-\alpha)\Delta W_{h_t} x_t \\ &= x_t + W_t x_t + (\alpha \Delta W_t + (1-\alpha)\Delta W_{h_t}) x_t \end{aligned}$$

Consequently, under the initial condition $x_1 = z_1$, the equivalence between the two systems can be demonstrated by the following set of equations:

$$\begin{cases} \alpha \Delta W_1 + (1-\alpha)\Delta W_{h_1} = M_1 \\ \alpha \Delta W_2 + (1-\alpha)\Delta W_{h_2} = M_2 \\ ... \\ \alpha \Delta W_L + (1-\alpha)\Delta W_{h_L} = M_L. \end{cases}$$

This admits the following matrix representation:

$$\begin{bmatrix} \alpha & 0 & 0 & \dots & 1-\alpha & 0 & \dots & 0 \\ 0 & \alpha & 0 & \dots & 0 & 1-\alpha & \dots & 0 \\ 0 & 0 & \alpha & \dots & 0 & 0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & 1-\alpha & 0 & 0 & \dots & \alpha \end{bmatrix} \begin{bmatrix} \Delta W_1 \\ \Delta W_2 \\ \Delta W_3 \\ \Delta W_4 \\ \Delta W_5 \\ \Delta W_6 \\ \vdots \\ \Delta W_L \end{bmatrix} = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6 \\ \vdots \\ M_L \end{bmatrix}.$$

Denote

$$A = \begin{bmatrix} \alpha & 0 & 0 & \dots & 1-\alpha & 0 & \dots & 0 \\ 0 & \alpha & 0 & \dots & 0 & 1-\alpha & \dots & 0 \\ 0 & 0 & \alpha & \dots & 0 & 0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & 1-\alpha & 0 & 0 & \dots & \alpha \end{bmatrix},$$

where $A = \alpha I + (1-\alpha)B$, where $I \in \mathbb{R}^{Ld \times Ld}$ is the identity matrix, and $B \in \mathbb{R}^{Ld \times Ld}$ is a matrix with exactly Ld entries equal to 1 (with positions determined by W_{h_t}), and all other entries equal to 0.

Next, we prove that $\alpha > \frac{1}{2}$ ensures $\det(A) \neq 0$. Suppose, for contradiction, that there exists a nonzero vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_{Ld} \end{bmatrix} \neq \mathbf{0},$$

such that

$$A\mathbf{x} = 0.$$

Then,

$$(\alpha I + (1-\alpha)B)\mathbf{x} = 0 \quad \Longleftrightarrow \quad B\mathbf{x} = -\frac{\alpha}{1-\alpha}\mathbf{x}.$$

Writing this component-wise, we obtain

$$\begin{bmatrix} x_{i_1} \\ x_{i_2} \\ \vdots \\ x_{i_{Ld}} \end{bmatrix} = \frac{\alpha}{1-\alpha} \begin{bmatrix} -x_1 \\ -x_2 \\ \vdots \\ -x_{Ld} \end{bmatrix},$$

where $\{i_1, i_2, \dots, i_{Ld}\} \subseteq \{1, 2, \dots, Ld\}$.

Taking the maximum absolute value on both sides yields

$$\max_{1 \le j \le Ld} |x_{i_j}| = \frac{\alpha}{1-\alpha} \max_{1 \le j \le Ld} |x_j|.$$

Since $\alpha > \frac{1}{2}$ implies $\frac{\alpha}{1-\alpha} > 1$, we obtain

$$\max_{1 \le j \le Ld} |x_{i_j}| > \max_{1 \le j \le Ld} |x_j|,$$

which is a contradiction. Therefore, no such nonzero $\mathbf{x}$ exists, and hence $\det(A) \neq 0$. Consequently, there exists a *unique* solution $\{\Delta W_k\}_{k=1}^{L}$ that establishes representational equivalence between the generalized state-based formulation and MoLEx.

In the nonlinear case, we show that the equivalence holds approximately when the adapters $\{\Delta W'_k\}_{k=1}^{L}$ are low-norm perturbations, using local linearization around the frozen pretrained path. Assume that each activation function f_k ($k = 1, \dots, L$) is twice continuously differentiable, with

- Lipschitz constant $L_f < \infty$ (uniform over k): $\|f_k(a) - f_k(b)\| \le L_f \|a - b\|$ for all a, b,
- Bounded Hessian norm $H < \infty$ (uniform over k): $\|\nabla^2 f_k(z)\| \le H$ for all z (operator norm).
- Assume the MoLEx adapters are small: $\|\Delta W'_k\| \le \xi$ with $\xi > 0, \forall\, k = \overline{1, L}$. Assume the matrix A above is invertible. Assume bounded intermediate states $\mathbb{E}[\|x_t\|^2] \le B < \infty$ and *pretrained mixing closeness*:

  $$\mathbb{E}\left[\left\|f_{h_t}(W_{h_t}x_t) - f_t(W_t x_t)\right\|^2\right] \le \gamma^2 \mathbb{E}[\|x_t\|^2] + \beta^2$$

  for small $\gamma, \beta \ge 0$ independent of ξ.

These conditions are satisfied by common activation functions (ReLU, GeLU, SwiGLU, etc.) on bounded domains or under mild regularization. Furthermore, the final assumption is well-motivated: recent work (He et al., 2024) shows that the outputs of Transformer blocks exhibit high similarity, which has driven further investigation into model pruning. Motivated by this observation, we impose a bound on the block outputs, referred to as the "*pretrained mixing closeness*" assumption.

We construct the state-based adapters $\{\Delta W_k\}_{k=1}^{L}$ exactly as in the linear case: solve the invertible linear system derived above (possible since $\alpha > \frac{1}{2}$) to match the *first-order* effective updates assuming identity Jacobians.

Consider a single update step t. By the Taylor theorem with integral remainder, for any layer k,

$$\begin{aligned} f_k((W_k + \Delta W_k')z_t) &= f_k(W_k z_t) + \nabla f_k(W_k z_t)\,(\Delta W_k' z_t) \\ &\quad + \int_0^1 (1-s)\nabla^2 f_k(W_k z_t + s\Delta W_k' z_t)\,ds\,(\Delta W_k' z_t \otimes \Delta W_k' z_t) \\ &= f_k(W_k z_t) + \nabla f_k(W_k z_t)\,(\Delta W_k' z_t) + R_k(z_t), \end{aligned} \tag{15}$$

where $\otimes$ denotes the appropriate bilinear action of the Hessian.

The remainder $R_k(z_t)$ satisfies

$$\|R_k(z_t)\| \le \frac{H}{2}\|\Delta W_k' z_t\|^2 \le \frac{H}{2}\xi^2\|z_t\|^2.$$

Substituting into the MoLEx update, we obtain

$$\begin{aligned} z_{t+1} = z_t &+ \alpha\Big[f_t(W_t z_t) + \nabla f_t(W_t z_t)(\Delta W_t' z_t) + R_t(z_t)\Big] \\ &+ (1-\alpha)\Big[f_{h_t}(W_{h_t} z_t) + \nabla f_{h_t}(W_{h_t} z_t)(\Delta W_{h_t}' z_t) + R_{h_t}(z_t)\Big]. \end{aligned} \tag{16}$$

Recall that the generalize state-based update is presented by

$$x_{t+1} = x_t + f_t(W_t x_t) + \alpha\,\Delta W_t x_t + (1-\alpha)\,\Delta W_{h_t} x_t.$$

Define the *linearized MoLEx control* at step t as

$$\tilde{M}_t := \alpha\,\nabla f_t(W_t z_t)\,\Delta W_t' + (1-\alpha)\,\nabla f_{h_t}(W_{h_t} z_t)\,\Delta W_{h_t}'.$$

The MoLEx update expands to

$$\begin{aligned} z_{t+1} &= z_t + \alpha f_t(W_t z_t) + (1-\alpha) f_{h_t}(W_{h_t} z_t) + \tilde{M}_t z_t + \alpha R_t(z_t) + (1-\alpha) R_{h_t}(z_t) \\ &= z_t + f_t(W_t z_t) + (1-\alpha)[f_{h_t}(W_{h_t} z_t) - f_t(W_t z_t)] + \tilde{M}_t z_t + \alpha R_t(z_t) + (1-\alpha) R_{h_t}(z_t). \end{aligned} \tag{17}$$

Assume that in step t, we have $z_t = x_t$, then the difference $z_{t+1} - x_{t+1}$ consists of three terms:

1. Base mixing error: $(1-\alpha)[f_{h_t}(W_{h_t} x_t) - f_t(W_t x_t)]$, with L^2 norm $\le |1-\alpha|^2(\gamma^2 B + \beta^2)$.
2. Control mismatch: $\tilde{M}_t x_t - (\alpha\Delta W_t + (1-\alpha)\Delta W_{h_t})x_t$. By linear-case construction and $\|\nabla f_k\| \le L_f$, this is $O(\xi\sqrt{B})$.
3. Remainder: $\|\alpha R_t + (1-\alpha)R_{h_t}\| = O(\xi^2 B)$.

Applying the Cauchy-Schwarz inequality $\|\sum_{i=1}^3 a_i\|^2 \le 3\sum\|a_i\|^2$ over the three terms, taking expectations, and absorbing higher-order terms yields the per-step bound. Then, for each t, we have

$$\mathbb{E}\left[\|z_{t+1} - x_{t+1}\|^2\right] \le C\Big(|1-\alpha|^2(\gamma^2 + \beta^2/B) + \xi^2\Big),$$

with constant $C > 0$ depending on α, ξ, L_f, H, B. As $\xi \to 0$, and $\gamma, \beta \to 0$ (or $|1-\alpha| \to 0$), the trajectories converge uniformly in L^2:

$$\max_t \mathbb{E}\left[\|z_t - x_t\|^2\right] \to 0.$$

Accordingly, a generalized state-based cross-layer communication mechanism suffices to approximate the MoLEx representation while achieving a markedly reduced computational cost. This completes the proof. □

A.2. Proposition 4.2

Proof. We analyze the forward dynamics by bounding the growth of hidden state norms across layers. The key idea is to show that MoC's adaptive routing mechanism can select control modules with smaller operator norms, leading to improved stability. For Parallel Control with $\alpha = 1$ (no routing), the update is:

$$x_{t+1}^c = x_t^c + f_t(W_t x_t^c) + \Delta W_t x_t^c. \tag{18}$$

Taking norms and applying the triangle inequality:

$$\begin{aligned} \|x_{t+1}^c\| &\leq \|x_t^c\| + \|f_t(W_t x_t^c)\| + \|\Delta W_t x_t^c\| \\ &\leq \|x_t^c\| + L_f \|x_t^c\| + \|\Delta W_t\| \|x_t^c\|. \end{aligned} \tag{19}$$

where we used the assumption that $\|f_t(W_t x_t^c)\| \leq L_f \|x_t^c\|$ (bounded nonlinearity with Lipschitz constant L_f).

Using the low-rank factorization $\Delta W_t = B_t A_t$ with $\|B_t\| \leq \beta_t$ and $\|A_t\| \leq \gamma$:

$$\|x_{t+1}^c\| \leq (1 + L_f + \beta_t \gamma) \|x_t^c\|. \tag{20}$$

By taking the expectation over data distribution, we have:

$$\mathbb{E}[\|x_{t+1}^c\| / \|x_t^c\|] \leq (1 + L_f + \beta_t \gamma) \tag{21}$$

Similarly, for the MoC dynamic with mixing parameter $\alpha \in [0, 1]$ and TopK routing, we imply:

$$x_{t+1} = x_t + f_t(W_t x_t) + \alpha \Delta W_t x_t + (1 - \alpha) \sum_{k \in S_K(x_t)} \pi_k(x_t) \Delta W_k x_t \tag{22}$$

where $S_K(x_t) \subseteq \{1, \ldots, L\}$ are the top-K selected experts and $\sum_{k \in S_K} \pi_k(x_t) = 1$. Similarly to Parallel Control, we take norms for MoC:

$$\begin{aligned} \|x_{t+1}\| &\leq \|x_t\| + \|f_t(W_t x_t)\| + \alpha \|\Delta W_t x_t\| + (1 - \alpha) \left\| \sum_{k \in S_K} \pi_k(x_t) \Delta W_k x_t \right\| \\ &\leq \|x_t\| + L_f \|x_t\| + \alpha \beta_t \gamma \|x_t\| + (1 - \alpha) \sum_{k \in S_K} \pi_k(x_t) \|\Delta W_k x_t\| \end{aligned} \tag{23}$$

where the last step uses convexity of the norm: $\| \sum_k \pi_k v_k \| \leq \sum_k \pi_k \|v_k\|$ for $\pi_k \geq 0, \sum_k \pi_k = 1$. Using $\|\Delta W_k\| \leq \beta_k \gamma$, we imply:

$$\begin{aligned} \|x_{t+1}\| &\leq \|x_t\| + L_f \|x_t\| + \alpha \beta_t \gamma \|x_t\| + (1 - \alpha) \sum_{k \in S_K} \pi_k(x_t) \beta_k \gamma \|x_t\| \\ &= (1 + L_f + \alpha \beta_t \gamma + (1 - \alpha) \beta_K(x_t) \gamma) \, \|x_t\| \end{aligned} \tag{24}$$

where we define the effective routing norm:

$$\beta_K(x_t) := \sum_{k \in S_K(x_t)} \pi_k(x_t) \beta_k. \tag{25}$$

Taking expectations over the data distribution $\mathcal{D}$:

$$\mathbb{E}[\|x_{t+1}\| / \|x_t\|] \leq (1 + L_f + \alpha \beta_t \gamma + (1 - \alpha) \mathbb{E}[\beta_K(x_t) | x_t] \gamma) \tag{26}$$

By routing learns to select experts with smaller operator norms on average:

$$\mathbb{E}[\beta_K(x_t) | x_t] < \beta_t, \tag{27}$$

then the norm bound forward pass will be smaller than Parallel Control:

$$(1 + L_f + \alpha\beta_t\gamma + (1-\alpha)\mathbb{E}[\beta_K(x_t)|x_t]\gamma) < (1 + L_f + \alpha\beta_t\gamma + (1-\alpha)\beta_t\gamma) = (1 + L_f + \beta_t\gamma),$$

which imply that MoC achieves improved forward stability compared to conventional control when routing is effective. □

Noting that with the assumption $\mathbb{E}[\beta_K(x_t)|x_t] < \beta_t$, we employ the load-balancing loss (Fedus et al., 2022; Lepikhin et al., 2020) to encourage balanced token routing across experts. This promotes more effective utilization of high-capacity experts, thereby increasing the likelihood of selecting suitable experts that meet the assumption.

A.3. Corollary 4.3

Proof. We analyze how control affects gradient flow during backpropagation through the network.

The backward gradient through block t is computed via the chain rule:

$$\frac{\partial \mathcal{L}}{\partial x_t} = J_t^\top \frac{\partial \mathcal{L}}{\partial x_{t+1}}, \tag{28}$$

then for MoC, differentiating the update equation gives:

$$\begin{aligned} J_t &= \frac{\partial}{\partial x_t}\left[x_t + f_t(W_t x_t) + \alpha \Delta W_t x_t + (1-\alpha)\sum_{k\in S_K} \pi_k(x_t)\Delta W_k x_t\right] \\ &= I + \frac{\partial f_t(W_t x_t)}{\partial x_t} + \alpha \Delta W_t + (1-\alpha)\sum_{k\in S_K}\pi_k(x_t)\Delta W_k \\ &\quad + (1-\alpha)\sum_{k\in S_K}\frac{\partial \pi_k(x_t)}{\partial x_t}\otimes(\Delta W_k x_t) \end{aligned} \tag{29}$$

For simplicity, we focus on the dominant terms and assume routing weights are approximately constant locally (i.e., the gradient w.r.t. π_k is small), giving:

$$J_t \approx I + \frac{\partial f_t(W_t x_t)}{\partial x_t} + \alpha \Delta W_t + (1-\alpha)\sum_{k\in S_K}\pi_k(x_t)\Delta W_k \tag{30}$$

Then we take norms and using the Lipschitz property of f_t with constant L_f:

$$\begin{aligned} \|J_t\| &\le 1 + \left\|\frac{\partial f_t(W_t x_t)}{\partial x_t}\right\| + \alpha\|\Delta W_t\| + (1-\alpha)\left\|\sum_{k\in S_K}\pi_k(x_t)\Delta W_k\right\| \\ &\le 1 + L_f\|W_t\| + \alpha\|\Delta W_t\| + (1-\alpha)\sum_{k\in S_K}\pi_k(x_t)\|\Delta W_k\| \\ &\le 1 + L_f\|W_t\| + \alpha\beta_t\gamma + (1-\alpha)\beta_K(x_t)\gamma \end{aligned} \tag{31}$$

Under the assumptions of Proposition 4.2, we obtain:

$$1 + L_f\|W_t\| + \alpha\beta_t\gamma + (1-\alpha)\beta_K(x_t)\gamma < 1 + L_f\|W_t\| + \beta_t\gamma,$$

which indicates that the Jacobian norm of MoC is more tightly bounded than that of Parallel Control, leading to a more stable gradient flow. □

In addition to the stability analyses of the representation and its gradient dynamics, we assess the fidelity of the learned representations. Assume $\Delta W^* x$ denote the optimal control signal that yields task-optimal representations. Given a MoC signal $g_{\text{moc}}(x)$, then the expected control-approximation error over data distribution is defined as follows:

$$\bar{\epsilon}^2 := \mathbb{E}\left[\|g_{\text{moc}}(x) - \Delta W^* x\|^2\right]. \tag{32}$$

A.4. Proposition 4.4

Proof. Given input x, then control error of Parallel Control is:

$$\delta_t^c(x) = \Delta W_t x - \Delta W^* x, \tag{33}$$

then control update at block t of MoC can be written as follows:

$$g_{\text{moc}}(x) = \alpha \Delta W_t x + (1-\alpha) \sum_{k \in S_K(x)} \pi_k(x) \Delta W_k x. \tag{34}$$

Next, the control representation error of MoC is:

$$\begin{aligned}
\delta_t(x) &= g_{\text{moc}}(x) - \Delta W^* x \\
&= \alpha[\Delta W_t x - \Delta W^* x] + (1-\alpha) \sum_{k \in S_K(x)} \pi_k(x)[\Delta W_k x - \Delta W^* x] && (35)\\
&= \alpha \delta_t^c(x) + (1-\alpha) \sum_{k \in S_K(x)} \pi_k(x) \delta_k^c(x), && (36)
\end{aligned}$$

where we have $\sum_{k \in S_K(x)} \pi_k(x) = 1$, $\pi_k(x) \geq 0, \forall\, k = \overline{1, L}$. Similarly,

By convexity of $\|\cdot\|^2$ (Jensen's inequality):

$$\begin{aligned}
\|\delta_t(x)\|^2 &= \left\| \alpha \delta_t^c(x) + (1-\alpha) \sum_{k \in S_K(x)} \pi_k(x) \delta_k^c(x) \right\|^2 \\
&\leq \alpha \|\delta_t^c(x)\|^2 + (1-\alpha) \left\| \sum_{k \in S_K(x)} \pi_k(x) \delta_k^c(x) \right\|^2 && (37)\\
& && (38)
\end{aligned}$$

Taking expectation over $x \sim \mathcal{D}$, $t = \overline{1, L}$:

$$\begin{aligned}
\bar{\epsilon}^2 &= \mathbb{E}[\|\delta_t(x)\|^2] \\
&\leq \alpha \mathbb{E}[\|\delta_t^c(x)\|^2] + (1-\alpha) \mathbb{E}\left[\left\| \sum_{k \in S_K(x)} \pi_k(x) \delta_k^c(x) \right\|^2 \right] \\
&= \alpha \bar{\epsilon}_c^2 + (1-\alpha) \bar{\epsilon}_K^2 && (39)
\end{aligned}$$

This completes the proof. □

B. Experimental Details

This section provides comprehensive details about our experimental setup, including hyperparameter configurations, dataset specifications, and implementation details for all models evaluated in this work.

B.1. Dataset Details

To comprehensively evaluate the effectiveness of our proposed method, we conduct experiments across multiple benchmark datasets covering different NLP task categories.

Natural Language Understanding (NLU) For NLU tasks, we employ the GLUE benchmark (Wang et al., 2018), a well-established benchmark suite that includes datasets such as SST-2, MRPC, CoLA, QNLI, RTE, and STS-B. These tasks collectively evaluate model performance on grammaticality judgment, sentiment classification, semantic similarity, and textual entailment across domain distributions.

Natural Language Generation and Commonsense Reasoning (NLG) For evaluating NLG and commonsense reasoning capabilities, we utilize eight benchmark datasets: BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), Social IQa (Sap et al., 2019), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC-Easy/ Challenge (Clark et al., 2018), and OpenBookQA (Mihaylov et al., 2018). These datasets assess model understanding of complex reasoning, physical commonsense, and social dynamics.

B.2. Model Architecture and Training Configuration

Encoder-Only Models. We utilized some Encoder-only models as the model backbones, particularly RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2020), on different model size (base and large), incorporating LoRA, DoRA, MoLEx and Parallel Control methods as the baselines. The detailed hyperparameter configurations for BERT-based models are illustrated in Table 4.

Decoder-Only Models. We utilized some Large scale models as the backbones, including LLaMA2-7B (Touvron et al., 2023), LLaMA3-8B (Grattafiori et al., 2024), Mistral-7B Instruct (Jiang et al., 2023), Qwen2.5-14B Instruct (Qwen et al., 2025). These pretrained models are also finetuned across our methods as well as the baselines. The detailed hyperparameter configurations for the decoder-only models are illustrated in Table 5. For routing, we consider two variants: (1) a linear router and (2) a cosine router with minimal extra parameters via a low-dimensional projection, which expands the latent space for richer representations.

MoC Configurations. For efficient Transformer training, we adopt batch-level routing in the MoC framework. Instead of routing each token independently, which would result in token-specific control assignments and hinder efficient batched computation, we make a unified routing decision for all tokens in a batch. To obtain the batch-level TopK controls, we aggregate token-level gate outputs using either "mode" or "mean" aggregation. The mode strategy selects the most frequently activated controls across tokens, while the mean strategy averages token-level probability vectors and chooses the controls with the highest averaged probabilities. In our experiments, we use either mode- or mean-based batch routing. The load-balancing weight is selected from $[0.001, 0.02]$, and the mixing coefficient is set to $\alpha \in \{0.6, 0.7, \ldots, 0.9, 0.95\}$, following Theorem 4.1.

Table 4. Hyperparameter configurations for DeBERTa and RoBERTa models on GLUE tasks. Common settings include AdamW optimizer, linear learning rate schedule, warmup ratio of 0.06, and sequence length of 128 tokens.

Model	Dataset	SST-2	MRPC	CoLA	QNLI	RTE	STS-B
	Optimizer	AdamW					
	WarmUp Ratio	0.06					
	LR Schedule	Linear					
	Max Seq Length	128					
$\text{DeBERTa}_{\text{base}}$	Batch Size	128	128	64	256	128	128
	Epochs	50	30	80	25	80	40
	Learning Rate	5e-4	4e-4	3e-4	4e-4	4e-4	4e-4
$\text{DeBERTa}_{\text{large}}$	Batch Size	64	32	32	32	64	32
	Epochs	20	30	20	10	20	10
	Learning Rate	4e-4	3e-4	5e-4	3e-4	4e-4	3e-4
$\text{RoBERTa}_{\text{base}}$	Batch Size	128	128	64	256	128	128
	Epochs	50	30	80	25	80	40
	Learning Rate	5e-4	4e-4	3e-4	4e-4	4e-4	4e-4
$\text{RoBERTa}_{\text{large}}$	Batch Size	64	32	32	32	64	32
	Epochs	20	30	20	10	20	10
	Learning Rate	4e-4	3e-4	5e-4	3e-4	4e-4	3e-4

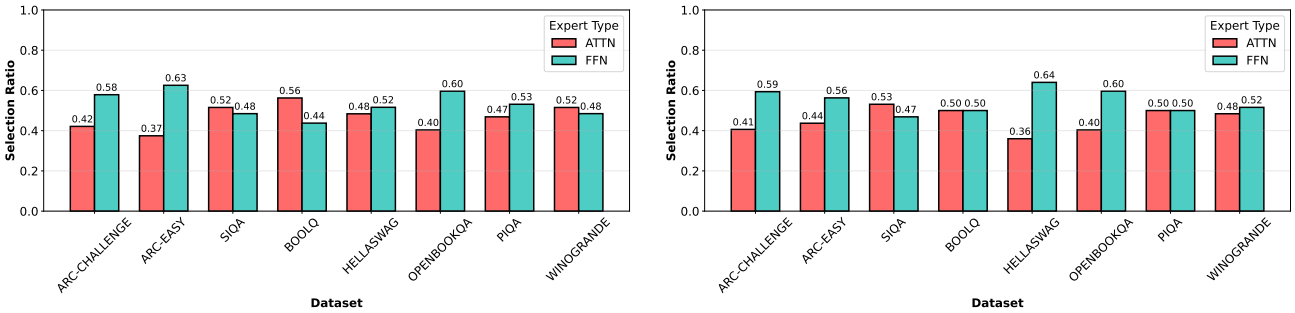
Compute Resources The compute resource used in our experiments consists of 8 NVIDIA H100 GPUs, each with 80GB of memory.

B.3. Additional Experiments

Inference Throughput. We measure the inference throughput of different LLMs across batch sizes, as shown in Figure 2. Given 100 samples, we run batched inference and compute throughput as the number of generated tokens divided by the

Table 5. Hyperparameter settings and Mixture-of-Control configuration for LLaMA2-7B, LLaMA3-8B, Mistralv3-7B, and Qwen2.5-14B. Settings are chosen to balance performance and training efficiency.

Parameter	**LLaMA2-7B**	**LLaMA3-8B**	**Mistralv3-7B**	**Qwen2.5-14B**
Learning rate	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}
Batch size	16	16	16	12
Gradient accumulation steps	2	2	2	2
Max sequence length	256	256	256	256
Projection dimension	256	256	256	256
Training epochs	3	3	3	3
Weight decay	0.01	0.01	0.01	0.01



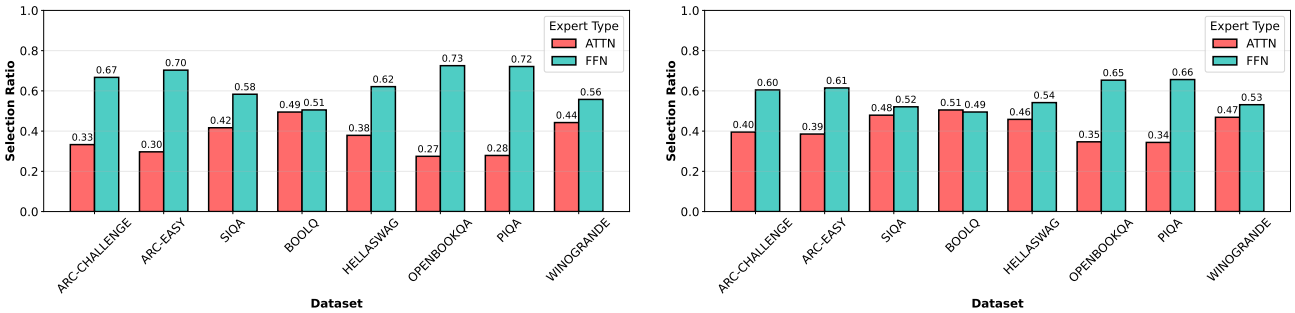
(a) Attention blocks

(b) FFN blocks

Figure 4. Cross-type routing behavior in LLaMA3-8B under the Top-$K = 1$ setting, conditioned on the local control being from (a) Attention blocks or (b) FFN blocks.

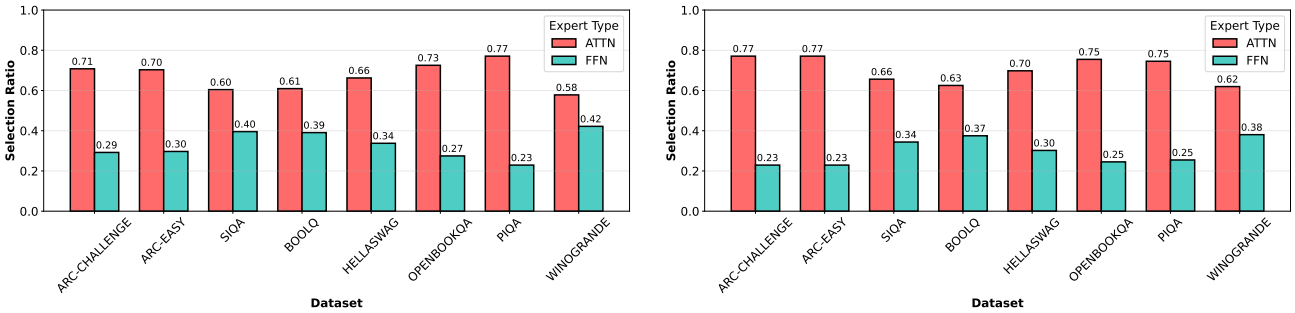
inference time for each batch. We then report the average throughput across all batches for each model.

Routing Behavior. To better understand the routing behavior of MoC, we visualize the selected experts across layers for both Attention and FFN blocks under different routing budgets. The routing patterns show that, at a given timestep, MoC is not limited to the local control associated with the current layer. Instead, it often selects experts from non-local layers, suggesting that the model learns to dynamically aggregate cross-layer information for richer intermediate representations. We report the visualizations for LLaMA3-8B under Top-1, Top-3, and Top-7 routing, with similar routing patterns observed across other backbone models (see Figures 4 to 12).



(a) Attention blocks (b) FFN blocks

Figure 5. Cross-type routing behavior in LLaMA3-8B under the Top-$K = 3$ setting, conditioned on the local control being from (a) Attention blocks or (b) FFN blocks.



(a) Attention blocks (b) FFN blocks

Figure 6. Cross-type routing behavior in LLaMA3-8B under the Top-$K = 7$ setting, conditioned on the local control being from (a) Attention blocks or (b) FFN blocks.

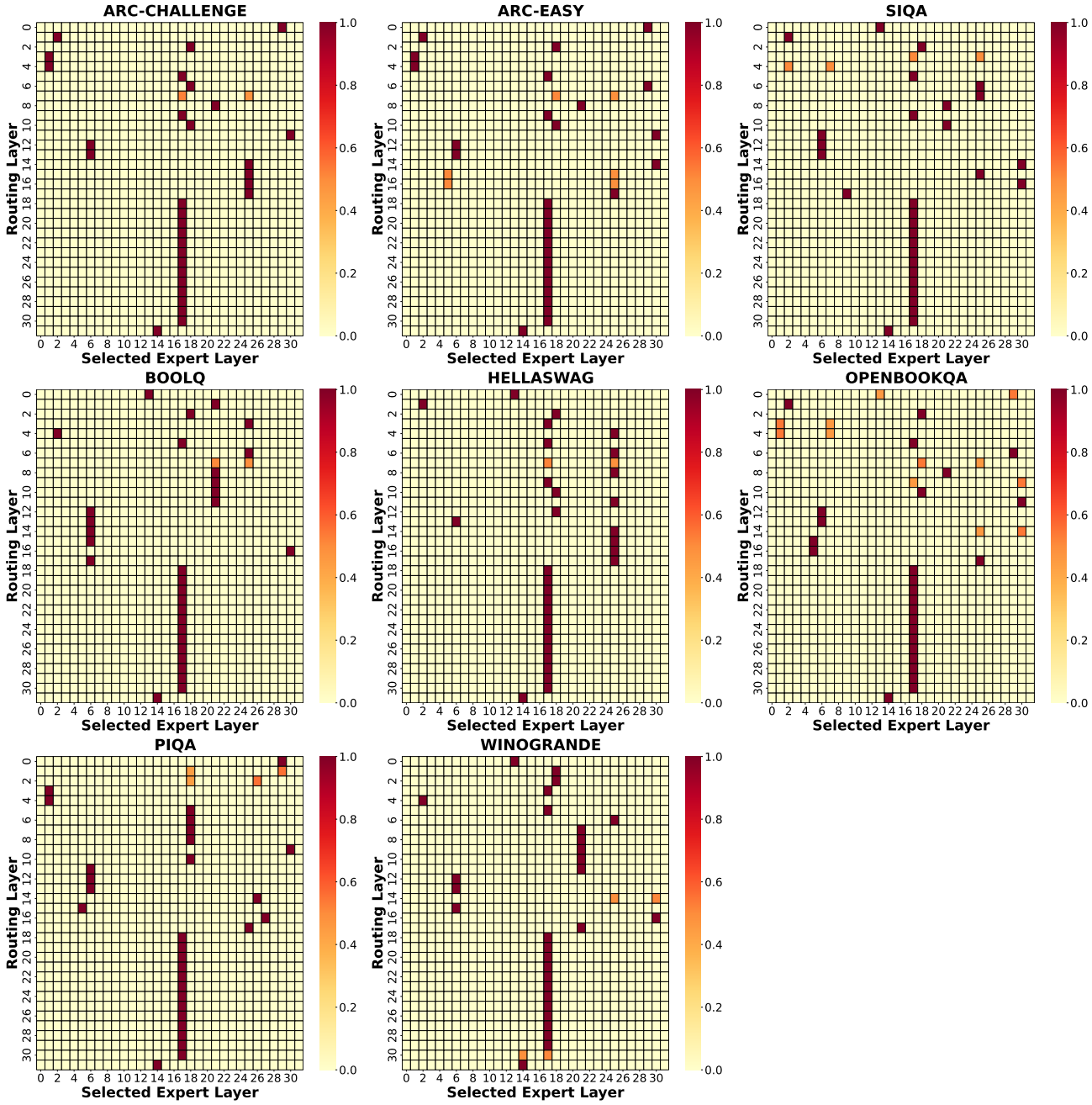


Figure 7. Routing behavior of MoC in LLaMA3-8B under Top-$K = 1$ routing when selecting Attention-block controls across layers.

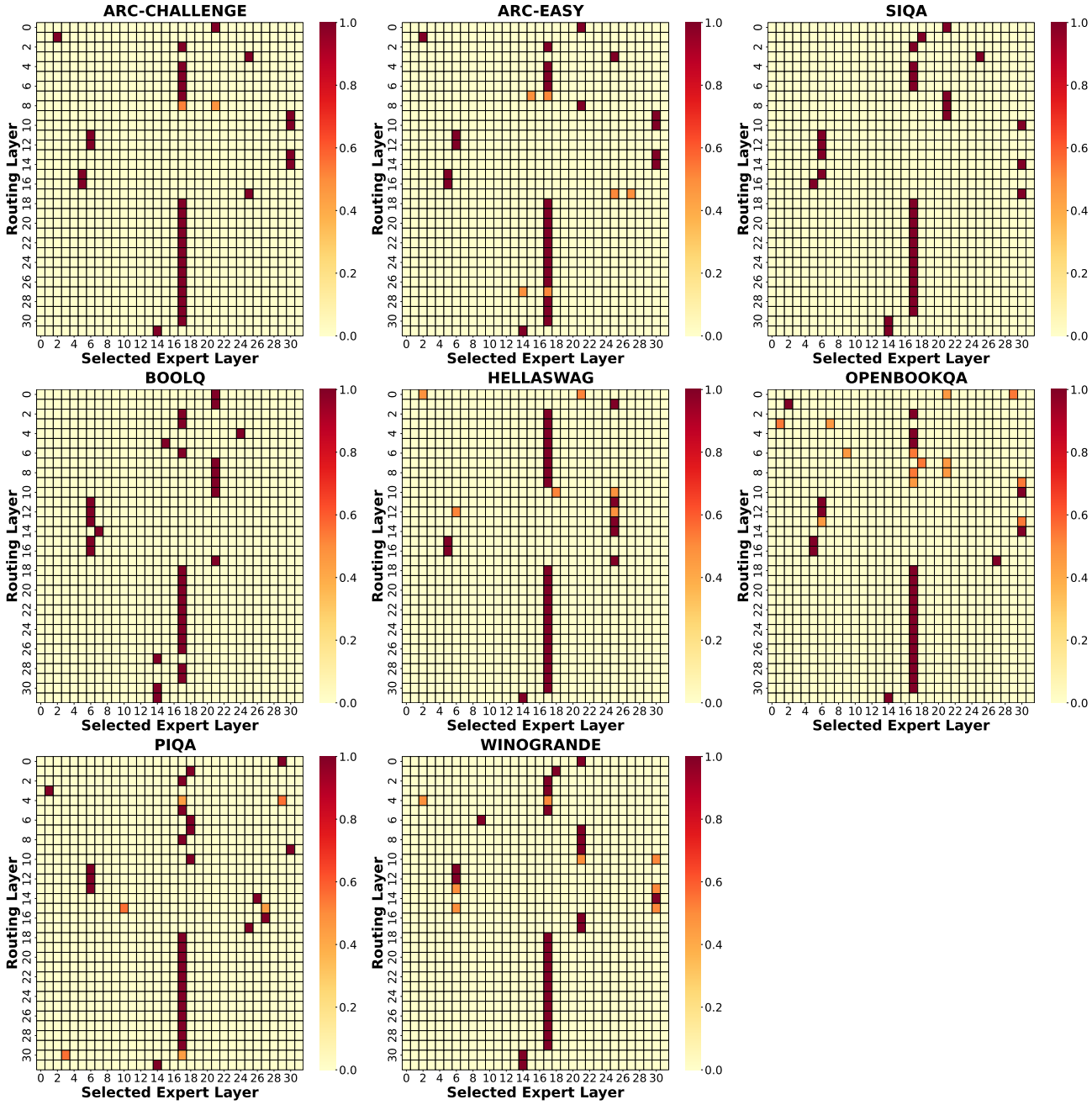


Figure 8. Routing behavior of MoC in LLaMA3-8B under Top-$K = 1$ routing when selecting FFN-block controls across layers.

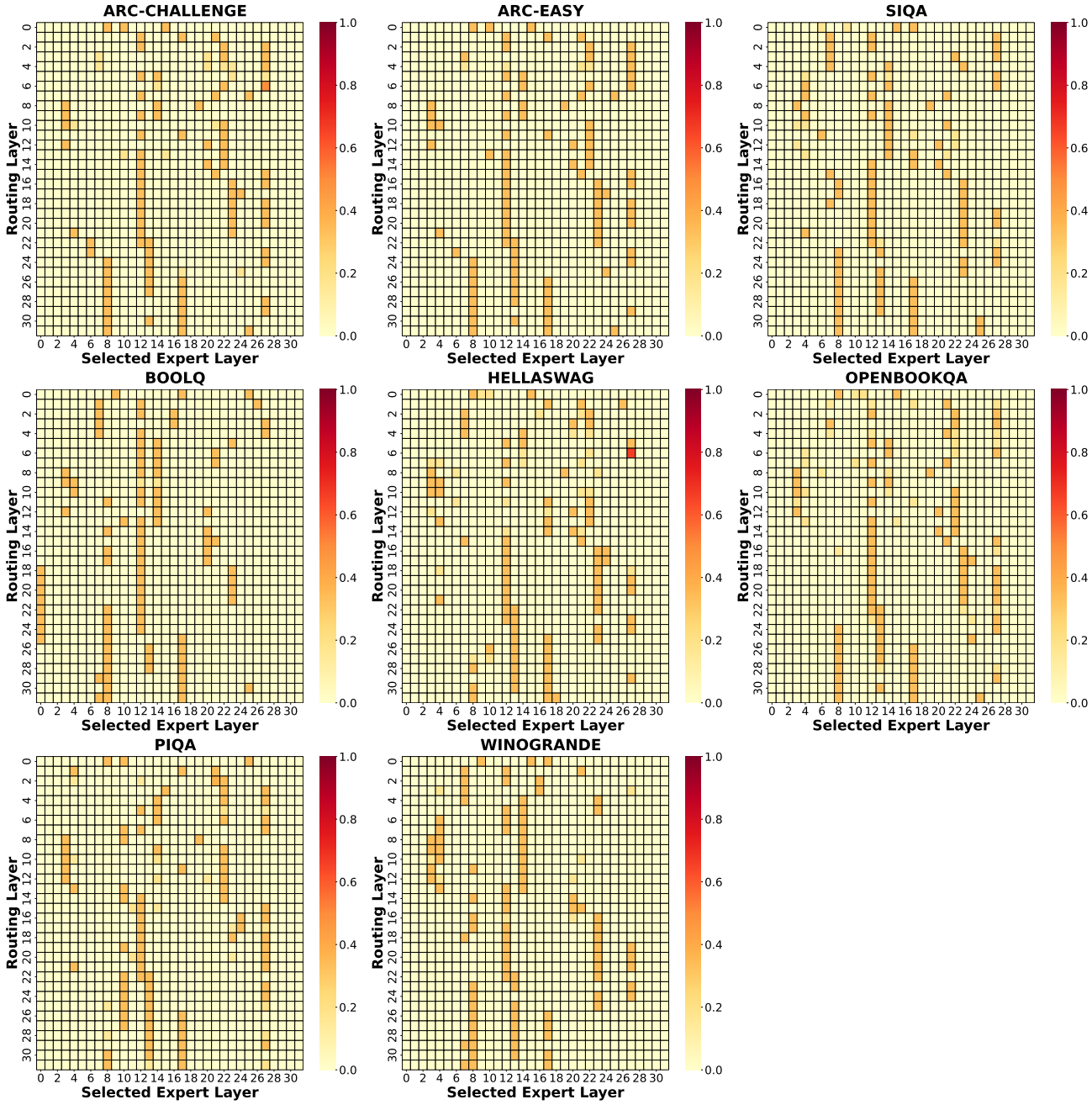


Figure 9. Routing behavior of MoC in LLaMA3-8B under Top-$K = 3$ routing when selecting Attention-block controls across layers.

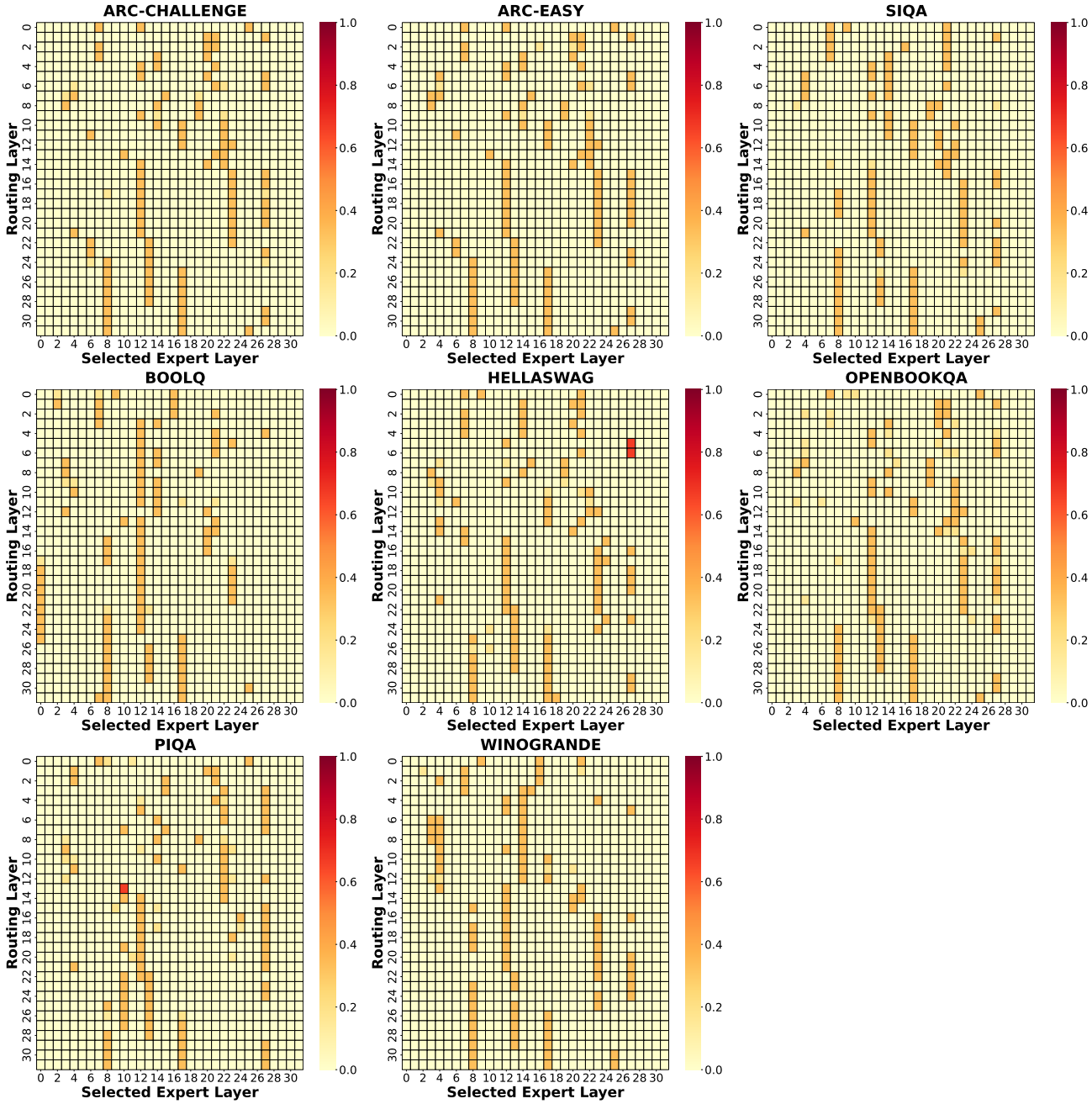


Figure 10. Routing behavior of MoC in LLaMA3-8B under Top-$K = 3$ routing when selecting FFN-block controls across layers.

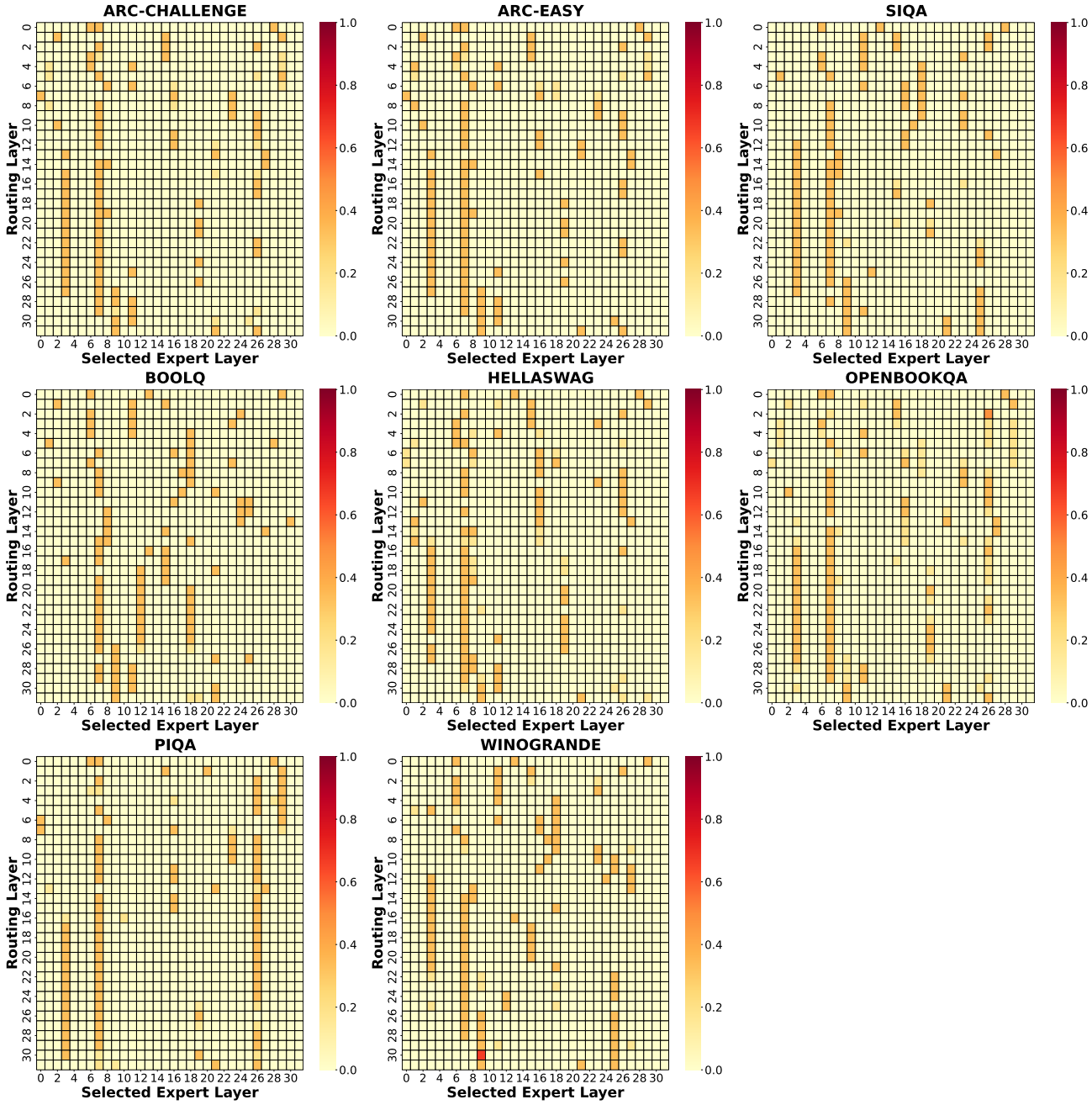


Figure 11. Routing behavior of MoC in LLaMA3-8B under Top-$K = 7$ routing when selecting Attention-block controls across layers.

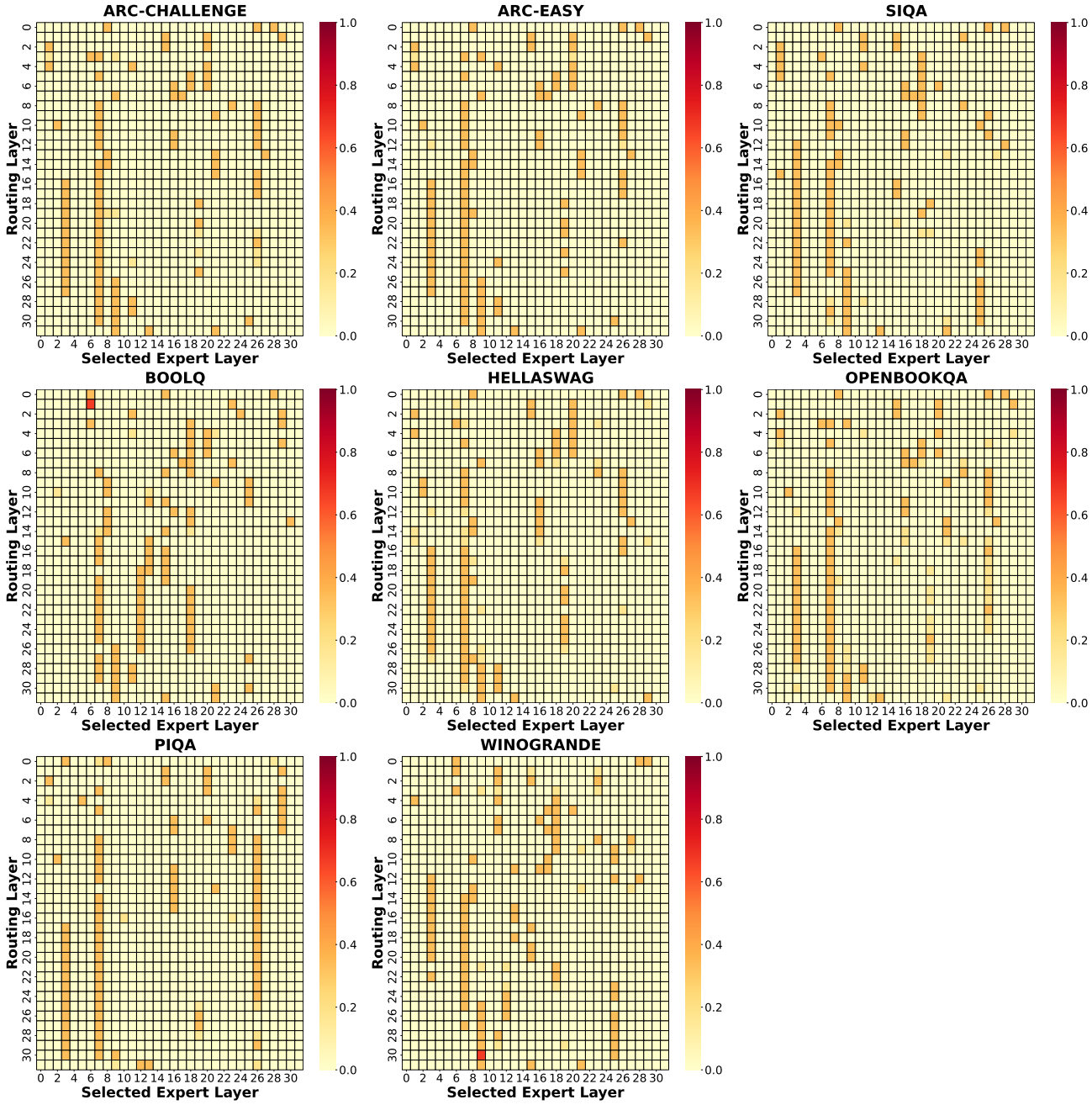


Figure 12. Routing behavior of MoC in LLaMA3-8B under Top-$K = 7$ routing when selecting FFN-block controls across layers.