A Layer-wise Analysis of Supervised Fine-Tuning

Qinghua Zhao[1], Xueling Gong[1], Xinyu Chen[1], Zhongfeng Kang[2], Xinlu Li[1]*
[1]Hefei University [2]Lanzhou University
{zhaoqh, xinlu.li}@hfuu.edu.cn kangzf@lzu.edu.cn
{gongxueling, chenxinyu}@stu.hfuu.edu.cn

Abstract

While critical for alignment, Supervised Fine-Tuning (SFT) incurs the risk of catastrophic forgetting, yet the layer-wise emergence of instruction-following capabilities remains elusive. We investigate this mechanism via a comprehensive analysis utilizing information-theoretic, geometric, and optimization metrics across model scales (1B-32B). Our experiments reveal a distinct depth-dependent pattern: middle layers (20%-80%) are stable, whereas final layers exhibit high sensitivity. Leveraging this insight, we propose Mid-Block Efficient Tuning, which selectively updates these critical intermediate layers. Empirically, our method outperforms standard LoRA up to 10.2% on GSM8K (OLMo2-7B) with reduced parameter overhead, demonstrating that effective alignment is architecturally localized rather than distributed. The code is publicly available at https://github.com/lshowway/base.

1 Introduction

Supervised Fine-Tuning has established itself as the cornerstone for aligning Large Language Models (LLMs) with human intent. Its efficacy is underscored by the observation that minimal supervision, as few as 1,000 curated examples, suffices to drastically transform base models into capable instruction-following agents (Zhou et al., 2023a; Ouyang et al., 2022).

Despite this empirical success, the mechanisms driving SFT remain nuanced. Research indicates that SFT primarily recalibrates attention patterns and shifts stylistic token distributions rather than altering underlying knowledge, effectively functioning as a "surface-level" adaptation (Wu et al., 2024; Lin et al., 2024; Wei et al., 2022). However, current parameter-efficient fine-tuning methods like LoRA ignore this depth-dependent heterogeneity. By applying updates uniformly across all layers (Ghosh et al., 2024), these methods operate under the suboptimal assumption that all layers contribute equally to alignment, potentially wasting parameter budget on insensitive layers.

A critical gap remains: while we know what changes during supervised fine-tuning, we have limited insight into where these changes occur across the model's depth and which layers are essential for instruction-following capabilities. Prior work has explored layer-wise knowledge localization (Meng et al., 2022) and layer-wise probing (Tenney et al., 2019), but these studies focused on what knowledge is stored where rather than where task adaptation occurs during fine-tuning. Understanding these layer-specific dynamics is crucial not only for theoretical insights but also for developing more efficient alignment procedures that concentrate computational resources where they matter most.

We conduct layer-wise analysis across models from 1B to 32B parameters, employing information-theoretic (entropy, effective rank), geometric (CKA, cosine similarity), and optimization (weight change) metrics. Through layer-wise probing, weight change tracking, and layer swapping and selective LoRA fine-tuning, we uncover a consistent depth-dependent adaptation pattern: 1) Representation similarity between Base and SFT models exhibits a progressive decline, culminating in a precipitous drop in the final layers; 2) Internal representations within both Base and SFT models undergo drastic divergence specifically in the upper layers; 3) Parameter update magnitudes mirror this trajectory, exhibiting significantly higher intensity in the top layers.

Based on this finding, we propose Mid-Block Efficient Tuning, which selectively updates only these critical intermediate layers. Experiments on mathematical reasoning (GSM8K) demonstrate that mid-block tuning achieves 37.5% accuracy, a 10-percentage-point improvement over standard

* Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9981–9992

LoRA (28%). The tendency is consistent across OLMo2-7B, 13B, 32B, and Mistral-7B, suggesting the pattern generalizes across model architectures and scales. Comparative studies confirm that targeting edge layers (bottom 20% or top 20%) results in performance degradation, validating the architectural locality of effective alignment. It is important to note that Mid-Block Efficient Tuning is not intended as a competing alternative to existing PEFT methods such as QLoRA or AdaLoRA. Rather, it serves as an analysis-driven proof-of-concept that validates our mechanistic findings regarding depth-dependent adaptation. We deliberately adopt standard LoRA as our primary baseline to isolate the effect of layer depth selection.

Drawing on these findings, we posit that supervised fine-tuning shares the same fundamental optimization dynamics as pre-training, i.e., updating parameters via backpropagation to encode new information, yet differs critically in data scale. We identify a functional divergence driven by these dynamics: the aggressive plasticity in the top layers causes incoming information to overwrite pre-existing features, marking these layers as the primary locus of catastrophic forgetting. Conversely, new information integrates with prior knowledge within the intermediate layers, which serve as the stable substrate for memory consolidation.

2 Related Work

We categorize related work into applications demonstrating the efficacy of SFT, and analytical studies probing the internal mechanisms behind these improvements.

2.1 Understanding SFT Effects

SFT functions as a critical mechanism for instruction alignment and generalization. Its efficacy is demonstrated across disparate tasks, including mathematical problem-solving (Wei et al., 2022; Chung et al., 2024; Tang et al., 2024), visual-language understanding (Liu et al., 2023), and suppressing toxic outputs and mitigating social biases (Huang et al., 2024).

However, SFT is constrained by data scarcity and knowledge boundaries, often incurring an "alignment tax" (Ouyang et al., 2022) where general capability gains are offset by catastrophic forgetting in specialized reasoning and increased hallucinations (Jiang et al., 2025; Ghosh et al., 2024). Beyond these performance trade-offs, the safety impact of SFT remains nuanced: rather than monotonically improving alignment, the process can paradoxically compromise pre-training guardrails or even exacerbate social biases relative to base models (Lyu et al., 2024; Itzhak et al., 2024).

2.2 Interpretability of Fine-tuning

A dominant perspective, the *Surface Alignment Hypothesis*, posits that SFT primarily elicits pre-trained capabilities rather than injecting new knowledge. This view is supported by observations that minimal data (e.g., 1K samples) suffices for robust alignment, suggesting SFT acts merely as a stylistic steer towards user-preferred formats (Zhou et al., 2023a; Jha et al., 2023; Kung and Peng, 2023). Such findings are corroborated by distributional analyses, which reveal that SFT induces shifts predominantly in stylistic tokens while preserving semantic representations (Lin et al., 2024).

From a mechanistic perspective, SFT recalibrates model behavior by sharpening attention on instruction tokens (Wu et al., 2024) and inducing representations in which closer layers between the base and SFT models share higher similarity (Rimsky et al., 2024). While this process enhances alignment with human cognition (Aw et al., 2024), it respects inherent knowledge boundaries: SFT optimizes response confidence rather than altering factual storage locations (Ren and Sutherland, 2025; Du et al., 2025). Consequently, forcing knowledge injection during this stage disrupts internal consistency and exacerbates hallucinations (Ren et al., 2024; Ghosh et al., 2024). Furthermore, feature localization via sparse autoencoders identifies the final layers as the critical locus for driving instruction adherence (He et al., 2025).

3 Methodology

3.1 Preliminaries

To deconstruct the layer-wise evolutionary dynamics induced by supervised fine-tuning, we first establish the notation for model representations and the spectral analysis framework. We analyze the internal representation space of a Base model $\mathcal{M}_b$ and its SFT counterpart $\mathcal{M}_s$. Both models share an identical architecture with L layers. Let $h^{(l)} \in \mathbb{R}^{T \times D}$ denote the hidden states at layer l for an input sequence of length T, where D is the hidden dimension. Given a dataset $\mathcal{D} = \{x_i\}_{i=1}^{N}$, we construct two distinct types of representation matrices to capture both local and global dynamics.

First, the *token-level* $H_i^{(l)} \in \mathbb{R}^{T\times D}$ is composed of the representations of all tokens within the i-th sample, which preserves fine-grained sequential details. Second, the *dataset-level* $\overline{H}^{(l)} \in \mathbb{R}^{N\times D}$ is constructed by stacking the mean-pooled vectors $\tilde{h}_i^{(l)} = \frac{1}{T}\sum_{t=1}^{T} h_i^{(l)}[t]$ from all N samples, representing the global data manifold.

Our analytical framework is grounded in the spectral properties of the Gram Matrix. For a generic representation matrix $Z \in \mathbb{R}^{T\times D}$ for a token-level $H_i^{(l)}$ or $Z \in \mathbb{R}^{N\times D}$ a dataset-level $\overline{H}^{(l)}$), the Gram matrix is defined as $K = ZZ^\top$, where the entry K_{jk} captures the pairwise similarity between elements. To quantify the intrinsic information capacity and dimensionality of the representation space, we employ the α-order matrix-based entropy $S_\alpha(K) = \frac{1}{1-\alpha}\log_2 \text{tr}\left(\left(\frac{K}{\text{tr}(K)}\right)^\alpha\right) = \frac{1}{1-\alpha}\log_2\left(\sum_j \tilde{\lambda}_j^\alpha\right)$, where $\tilde{\lambda}_j$ are the eigenvalues of the normalized Gram matrix, forming a probability distribution such that $\sum \tilde{\lambda}_j = 1$. This unified definition allows us to rigorously measure whether SFT preserves, compresses, or displaces the information encoded in the pre-trained features.

3.2 Optimization Dynamics

While the metrics in subsequent sections characterize the SFTs in the representation manifold, the underlying mechanism originates in the optimization landscape. We posit that the intensity of parameter updates reflects the adaptation effort allocated to each layer. Let $\Theta^{(l)} = \{W_Q, W_K, W_V, W_O\}^{(l)}$ denote the set of all learnable projection matrices of attention module within the l-th Transformer layer, where W_Q, W_K, W_V, and W_O denote the query, key, value, and output projection weight matrices of the self-attention module, respectively. To quantify the magnitude of this re-optimization, we define the weights change metric $\Delta\mathcal{W}^{(l)}$ as the aggregated Frobenius distance between the parameter configurations of the SFT model ($\mathcal{M}_s$) and the Base model ($\mathcal{M}_b$):

$$\Delta\mathcal{W}^{(l)} = \sqrt{\sum_{W\in\Theta^{(l)}} ||W_s - W_b||_F^2} \tag{1}$$

A high value of $\Delta\mathcal{W}^{(l)}$ indicates that the layer has experienced aggressive parameter modifications induced by the supervised fine-tuning objective. Our central hypothesis is that the layers closest to the output loss function are forced to undergo the most significant structural changes to accommodate the new task constraints. This parameter-level reconstruction acts as the driving force, physically displacing the pre-trained weights and thereby causing the information compression and geometric restructuring observed in the representation space.

3.3 Information Dynamics

The parameter updates described in Section 3.2 inevitably alter the information capacity of the representation space. We employ the matrix-based entropy framework to monitor this SFT, specifically testing the information bottleneck hypothesis: that SFT forces the model to compress generic pre-training features (old information) to accommodate task-specific constraints (new information).

Prompt Entropy. To quantify the intra-sequence information density and determine whether SFT compresses fine-grained token details, we compute the average entropy over all N samples. For the i-th sample representation $H_i^{(l)}$, the prompt entropy is:

$$S_p^{(l)}(i) = S_\alpha(H_i^{(l)}(H_i^{(l)})^\top) \tag{2}$$

A decrease in $S_p^{(l)}$ ($\Delta < 0$) suggests that SFT induces token compression, filtering out redundant pre-training noise to focus on task-relevant signals.

Dataset Entropy. To assess the inter-sample diversity and verify if the strong supervision signal causes mode collapse, we compute the entropy of the dataset-level matrix $\overline{H}^{(l)}$:

$$\mathcal{S}_d^{(l)} = S_\alpha(\overline{H}^{(l)}(\overline{H}^{(l)})^\top) \tag{3}$$

A negative SFT $\Delta\mathcal{S}_d^{(l)} < 0$ indicates that SFT pulls sample representations closer to form tighter, task-specific manifolds.

Effective Rank and Deficiency. To gauge the true dimensionality of the representation space beyond simple rank constraints, we utilize the effective rank. Let $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r > 0$ be the singular values of the dataset matrix $\overline{H}^{(l)}$. We define the normalized singular value distribution as $p_j = \sigma_j^2 / \sum_k \sigma_k^2$. The effective rank is:

$$\text{EffRank}(\overline{H}^{(l)}) = \exp\left(-\sum_{j=1}^{r} p_j \log p_j\right) \tag{4}$$

We complement this with the algebraic rank deficiency, defined as $Def = \min(N, D) - |r|$. A

reduction in effective rank or an increase in deficiency signifies that the model is utilizing a smaller, more efficient subspace to encode the task, confirming the displacement of the broad pre-training basis.

Beyond spectral properties, we examine the raw encoding efficiency through Sparsity, which measures the fraction of inactive neurons ($|z| < \epsilon$) averaged over all dimensions. Higher sparsity indicates SFT is performing explicit feature selection, pruning irrelevant pre-training neurons to dedicate capacity solely to the target instruction logic.

3.4 Geometric Restructuring

Finally, we analyze the external manifestation of these internal SFTs. First, we analyze how the model navigates the semantic space across a sequence. Let $v_t = h_t^{(l)} - h_{t-1}^{(l)}$ be the difference vector between consecutive tokens. The curvature quantifies the smoothness of the reasoning path:

$$\mathcal{C}_i^{(l)} = \frac{1}{(T-2)\pi} \sum_{t=1}^{T-2} \arccos\left(\frac{v_t^\top v_{t+1}}{||v_t|| \cdot ||v_{t+1}||}\right) \tag{5}$$

where π is normalization factor, $\mathcal{C}^{(l)} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{C}_i^{(l)}$. A reduction in curvature ($\Delta\mathcal{C} < 0$) enables the model to maintain more coherent long-term dependencies by simplifying the route through the representation space.

To determine if SFT merely rotates the representation space (isomorphic transformation) or fundamentally restructures it, we employ Centered Kernel Alignment (CKA). Given $K_b = \overline{H}_b^{(l)}(\overline{H}_b^{(l)})^\top$ and $K_s = \overline{H}_s^{(l)}(\overline{H}_s^{(l)})^\top$ be the dataset-level Gram matrices. We first introduce the centering matrix $J = I_N - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$ to remove the mean component. The centered Gram matrices are $\tilde{K}_b = JK_bJ$ and $\tilde{K}_s = JK_sJ$. The CKA similarity is computed as:

$$\mathcal{A}_{CKA}^{(l)} = \frac{\langle \tilde{K}_b, \tilde{K}_s \rangle_F}{\|\tilde{K}_b\|_F \|\tilde{K}_s\|_F} = \frac{\mathrm{tr}(\tilde{K}_b\tilde{K}_s)}{\sqrt{\mathrm{tr}(\tilde{K}_b^2) \cdot \mathrm{tr}(\tilde{K}_s^2)}} \tag{6}$$

A value of $\mathcal{A}_{CKA}^{(l)} \ll 1$ signals that the original manifold structure has been substantially restructured to accommodate the new task-specific representations.

While CKA captures global structural similarity, we measure explicit directional reorientations using cosine similarity and mean shift. Let $\mu^{(l)}$ denote the centroid of the layer representations.

$$\mathcal{S}_{cos}^{(l)} = \frac{1}{N}\sum_{i=1}^{N} \frac{(\overline{h}_{b,i}^{(l)})^\top \overline{h}_{s,i}^{(l)}}{\|\overline{h}_{b,i}^{(l)}\|_2 \|\overline{h}_{s,i}^{(l)}\|_2}, \tag{7}$$
$$\mathcal{D}_{SFT}^{(l)} = \|\mu_s^{(l)} - \mu_b^{(l)}\|_2$$

A sharp decay in $\mathcal{S}_{cos}^{(l)}$ coupled with a spike in $\mathcal{D}_{SFT}^{(l)}$ provides the coordinate-level evidence that the representation has been physically transported to a new region of the vector space, driven by the optimization dynamics described in Section 3.2.

3.5 Evaluation Protocol

To ensure rigorous reproducibility and properly quantify the evolutionary gap between the Base and SFT models, we categorize our analysis into three computation modes based on the aggregation scope of the representations. Let f_k denote a specific metric function defined in the previous sections. We strictly distinguish between local token dynamics and global manifold properties.

- **Sample-Level Difference** For metrics quantifying local properties (e.g., prompt entropy, curvature, cosine similarity), we compute the change per sample and report the average. This captures the mean intra-sample evolutionary magnitude:

$$\Delta Q_s^{(l)} = \frac{1}{N}\sum_{i=1}^{N}(f_k(H_{s,i}^{(l)}) - f_k(H_{b,i}^{(l)})) \tag{8}$$

- **Dataset-Level Difference** For global properties dependent on the full distribution (e.g., dataset entropy, effective rank), we apply the metric to the holistic dataset matrix to prevent aggregation artifacts:

$$\Delta Q_d^{(l)} = f_k(\overline{H}_s^{(l)}) - f_k(\overline{H}_b^{(l)}) \tag{9}$$

- **Alignment Score** For interaction metrics that require dual inputs (e.g., CKA), we measure the geometric correspondence directly without subtraction:

$$A_a^{(l)} = f_k(\overline{H}_b^{(l)}, \overline{H}_s^{(l)}) \tag{10}$$

4 Experiments

In this section, we conduct a comprehensive analysis to understand the impact of SFT on model representations and leverage these insights to propose a more efficient tuning strategy.

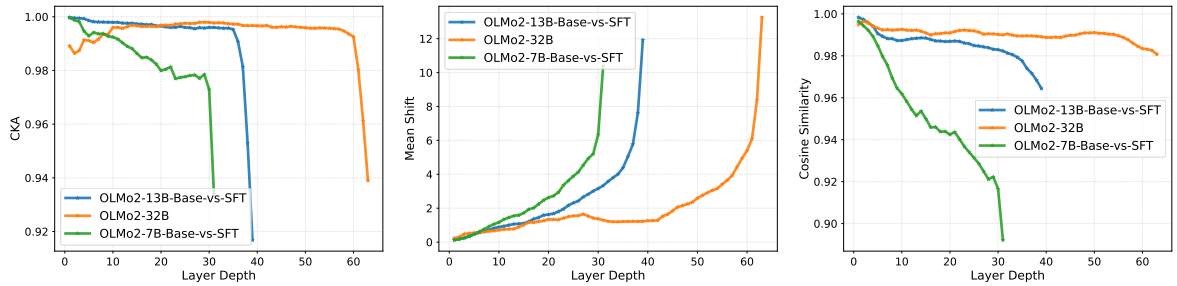

Figure 1: Representational divergence metrics (CKA, Cosine Similarity and Mean Shift) across layers.

4.1 Experimental Setup

Models We focus on model families that provide both pre-trained (Base) and SFT checkpoints. We utilize Mistral-7B and the OLMo2 series (scaling from 1B to 32B), to rigorously analyze the impact of SFT without the confounding factors of complex alignment techniques (e.g., RLHF or DPO). To further contextualize the model selection in our experimental setup, we provide a survey of commonly used open-source LLMs and their alignment pipelines, justifying why OLMo2 and Mistral-7B are the most suitable choices for isolating pure SFT dynamics. Details are provided in Appendix C. We conduct experiments using official checkpoints from the Hugging Face Hub to ensure reproducibility. Specifically, we evaluate mistralai/Mistral-7B-v0.1 (Base) and its Instruct-v0.1 (SFT) variant. For the OLMo 2 family, we utilize the entire suite including allenai/OLMo-2-0425-1B, allenai/OLMo-2-1124-7B, allenai/OLMo-2-1124-13B, and allenai/OLMo-2-0325-32B across both Base and SFT versions.

Datasets To ensure a comprehensive evaluation of model capabilities, we employ a diverse suite of benchmarks: Massive Multitask Language Understanding (MMLU, (Hendrycks et al., 2021)), Grade School Math 8K (GSM8K, (Cobbe et al., 2021)), WikiText Language Modeling Dataset (WikiText, (Merity et al., 2017)), Instruction Following Evaluation (IFEval, (Zhou et al., 2023b)), HumanEval (Chen et al., 2021), Multi-Turn Benchmark (MT-Bench, Zheng et al. 2023), and ToxiGen (Hartvigsen et al., 2022). We categorize these into two analysis granularities. For token-level analysis (MMLU, GSM8K, WikiText, and IFEval), where representations are extracted for every token in the sequence, we sample 100 instances from the test sets. Conversely, for pooled-level analysis (including all datasets), we adopt a strategy where each input corresponds to a single vector, using a sample size of 1,000 instances from the respective test sets.

Implementation Details. Given the high dimensionality of representations (e.g., $L = 40, T = 256, D = 4096$), storing full representation tensors is computationally infeasible. We thus employ streaming computation to process representations on-the-fly, minimizing memory overhead. To ensure deterministic evaluation, particularly for invariance metrics involving augmentations, we fix global random seeds and strictly enforce sample processing order. Furthermore, for entropy-based metrics, to ensure comparability across layers with varying dimensions, we report the normalized matrix entropy: $\bar{S}_\alpha(K) = S_\alpha(K)/\log_2(N)$, where N is the batch size. $\epsilon = 0.01$ is set to compute representation sparsity. The code is publicly available at https://anonymous.4open.science/r/base_sft.

4.2 Layer-wise Dynamics of SFT

To deconstruct the evolutionary trajectory induced by supervised fine-tuning, we first investigate how SFT alters the internal representations compared to the Base model across different layers.

Interact-Divergence We quantify the divergence between the Base and SFT models using Cosine Similarity, CKA and mean shift. As illustrated in Figure 1, distinct evolutionary phases were observed across model scales. Taking OLMo2-32B as an example, the CKA remains stable in the shallow layers (0–56), where the score plateau stays above 0.98. However, this was followed by a sharp drop in deeper layers (56 to the end), where the score decays significantly, dropping to approximately 0.94 in the final layer. This directional shift was corroborated by the mean shift (and also cosine similarity) metric, which remains negligible (< 1.0) for the majority of the shallow layers (35 vs. 40 layers)

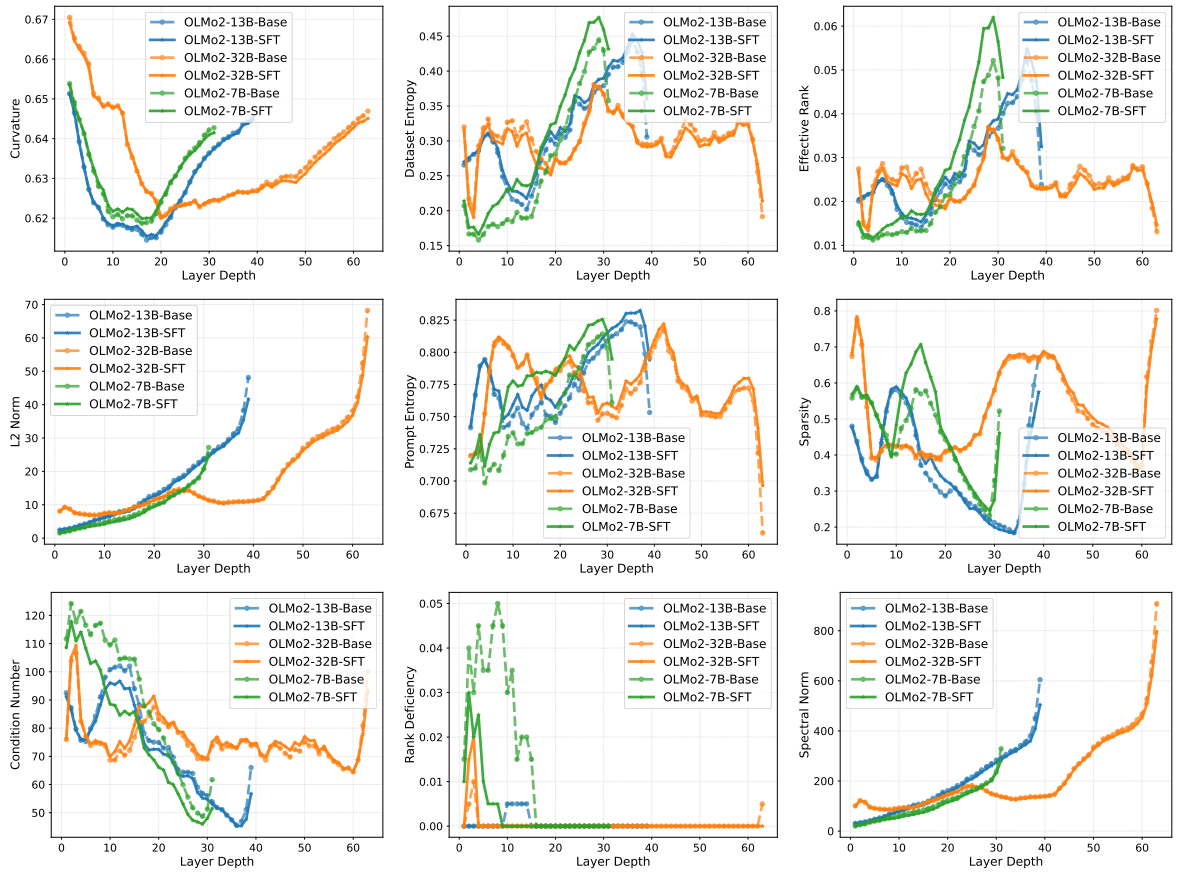

Figure 2: Intrinsic representation metrics across layers for Base and SFT models.

but spikes exponentially in the final 5 layers, reaching a magnitude of over 12.0. To verify that these SFTs are induced by supervised fine-tuning rather than random initialization noise, we conducted a robustness check using different random seeds. A *t*-test confirms the statistical significance of these differences ($p < 0.01$).

Independent-Convergence While the previous analysis highlighted the significant divergence between Base and SFT representations, we examine the layer-wise trends of intrinsic metrics (e.g., effective rank) for both Base and SFT models, as shown in Figure 2. In contrast to the interact-divergence observed in Figure 1, the evolution of Base and SFT models remains remarkably synchronized. Specifically, the curves exhibit a consistent three-stage pattern: an initial transition occurs from the embedding layer (Layer 0) to Layer 1, followed by a relatively stable phase across the extensive intermediate blocks. In this middle zone, the geometry reflects a semantic expansion, where effective rank plateaus at its peak (e.g., ~ 0.05 for OLMo2-13B) and the Condition Number drops to a noise-resilient trough of ~ 45, physically functioning as a high-dimensional, stable substrate for reasoning. Finally, a distinct functional shift re-emerges within the last $\sim 20\%$ layers, manifesting as an information bottleneck. Here, the effective rank collapses to < 0.01 while Spectral Norm explodes to over 500, forcefully compressing features into a low-rank, high-magnitude state to drive the decisive output distribution.

4.3 Locating the Task Adaptation

The SFT observed in Section 4.2 prompts us to investigate this through probing, weights analysis, and layer swapping.

Layer-wise Probing We hypothesize that the significant representational SFTs in later layers correspond to the emergence of task adaptation capabilities. To verify this, we perform a layer-wise probing experiment where we use the output of each intermediate layer to directly predict the next token. Figure 3 displayed the next token prediction accuracy across layers. The results reveal a striking "dormancy-to-emergence" pattern, which is particularly pronounced in larger scales. Taking the OLMo2-32B model as a prime example, the

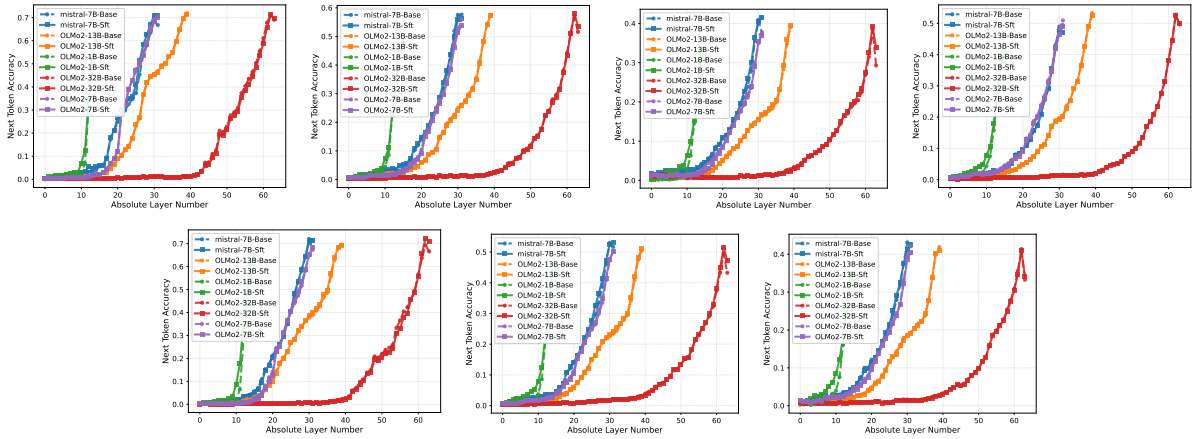

Figure 3: Next-token prediction accuracy of layer-wise probing, from left to right is GSM8K, MMLU, IFEval, WikiText, HumanEval, MT-Bench, and ToxiGen dataset.

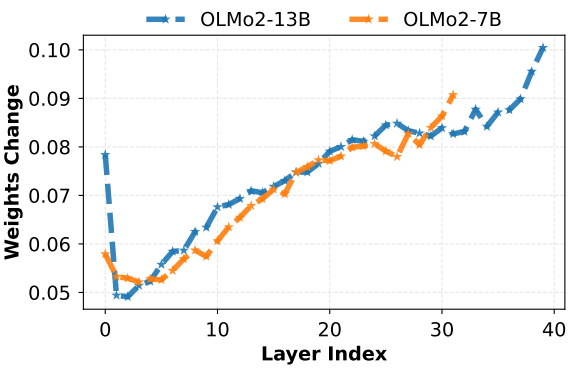

Figure 4: Layer-wise magnitude of weight updates (L_2 norm).

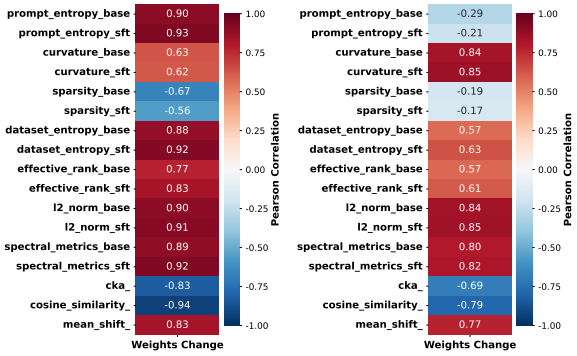

Figure 5: Correlation between layer-wise magnitude of weight updates and representation metrics, OLMo2-7B (left), OLMo2-13B (right).

probing accuracy remains negligible (below 0.05) for the vast majority of its depth (the first 50 layers). However, a sharp phase transition occured in the final block: within the last 14 layers, the accuracy exhibits a vertical ascent, surging from near-zero to over 0.60 on the MMLU dataset. Although a slight decrease is observed in the final layer (from 0.60 to 0.52), we defer analysis of this phenomenon and focus on the transition from mid-to-late layers.

Layer-wise Weight Change To physically ground the representational shift observed in Section 4.2, we revisit the optimization dynamics. We fine-tuned the models on a distinct downstream train dataset and recorded the magnitude of parameter updates (L_2 norm of weight changes $\Delta\mathcal{W}^{(l)}$) for each layer. As shown in Figure 4, the update intensity was non-uniform. Taking the OLMo2-13B model as an example, the weight change exhibits a J-shaped trajectory: minimal ($\sim$0.05) in the early layers, it climbs monotonically after the midpoint, reaching a peak of over 0.10 in the final layer.

The correlation heatmap in Figure 5 provided statistical confirmation. Taking OLMo2-13B as an example, we observed a strong negative correlation ($r = -0.79$) between weights change and cosine similarity, and a strong positive correlation ($r > 0.8$) with spectral metrics. It maybe due to that, during SFT, the supervision signal from the loss function is strongest at the output and attenuates as it back-propagates. Consequently, the "knowledge" required for the new task is preferentially encoded in the late layers through aggressive weight updates, while the bottom layers, shielded by gradient attenuation, implicitly act as a frozen feature extractor.

Layer-wise Swapping To establish a causal link between layer groups and model performance, we conduct a layer swapping experiment by replacing specific blocks of layers in the Base model with their SFT counterparts (and vice versa). Figure

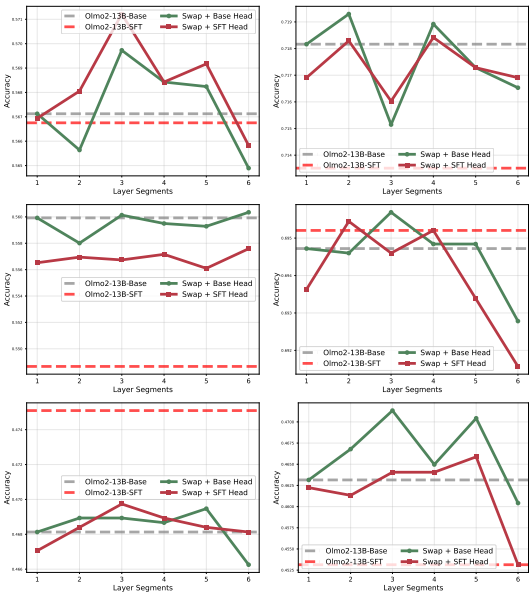

Figure 6: Model performance evaluation under block-wise layer swapping between Base and SFT weights, from left to right is MMLU, GSM8K, WikiText, HumanEval, MT-Bench, ToxiGen dataset.

6 reveals an inverted-U-shaped pattern: replacing layers at either early or late results in performance degradation, while replacing middle layers can occasionally lead to slight improvements. For instance, on the MMLU dataset using the base head of OLMo2-13B, replacing the first 20% or last 20% of layers causes performance drops of 0.001 and 0.002, respectively, in contrast, replacing middle layers produces a gain of 0.003. Despite the small magnitude of this improvement, the overall trend remains consistent across various datasets and models. We hypothesize that the marginal nature of these changes arises from the limited representational differences between the Base and SFT models, as illustrated in Figure 2.

This counterintuitive observation prompts us to consider the following explanations: 1) The later layers are more task-specific and exhibit stronger coupling with adjacent layers (as shown in Figure 2, where changes in the later layers are more pronounced). Consequently, indiscriminately replacing them leads to performance drops. In contrast, the middle layers have weaker coupling with the early and late layers, as they store more general knowledge (evidenced by the relatively lower accuracy in the early layers shown in Figure 3). The new knowledge introduced by SFT does not overwrite or displace the existing knowledge (i.e., no catastrophic forgetting occurs), instead, it may fully

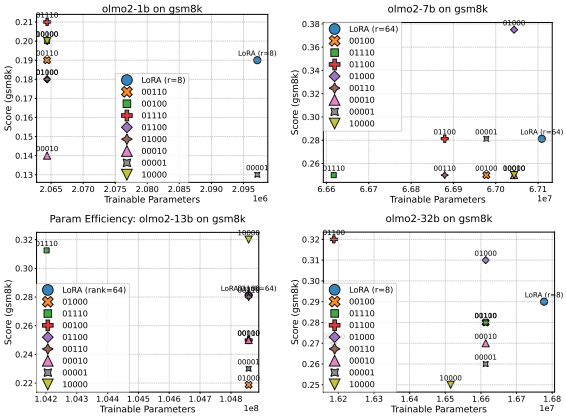

Figure 7: Mid-Block Efficient Tuning, from left to right is OLMo2-(1B, 7B, 13B, 32B), respectively.

retained and primarily resides in the middle layers.

4.4 Mid-Block Efficient Tuning

To test this, we divide the model layers into M equal segments and compare performance when the added LoRA parameters (with $M = 5$ and rank $\in \{8, 64\}$) are concentrated in different segments, while keeping the total number of trainable parameters nearly constant. Experiments are conducted on the OLMo2 family using the GSM8K and MMLU dataset, with GSM8K results shown in Figure 7.

For OLMo2-7B, targeting only the upper-middle segment (`01000`) yielded 0.375, substantially outperforming the full-layer LoRA baseline (0.28) by nearly 10 percentage points. Similarly, on OLMo2-32B, the mid-upper configuration (`01100`) achieved 0.32, surpassing the baseline (0.29) while using fewer trainable parameters. On OLMo2-13B, the wider middle configuration (`01110`) led with 0.30 versus the baseline's 0.27. In contrast, focusing solely on the bottom segment (`10000`) yielded poor results, for instance, 0.22 on OLMo2-13B (vs. baseline 0.27) and 0.25 on the 32B model (7 points below the best middle-segment setting). Tuning only the top segment (`00001`) performed equally poorly: on OLMo2-1B, it scores just 0.135 (vs. baseline 0.19), indicating that adapting only the output mapping without modifying internal reasoning is insufficient. Overall, the performance gap between the best middle-segment and worst edge-segment configurations frequently exceeds 20% of the total score, underscoring the critical importance of targeted layer selection in LoRA adaptation.

To further investigate the sensitivity of performance to finer-grained boundary choices, we conduct additional ablation studies with $M = 3$ and

$M = 10$ segments. The results consistently exhibit an inverted-U-shaped pattern across granularities, confirming that the intermediate region represents a broad and robust representational plateau rather than a sensitive peak. Details are provided in Appendix A. We further provide quantitative evidence for pre-training knowledge retention as detailed in Appendix B.

Conclusion

By probing the layer-wise dynamics of SFT from information-theoretic, geometric, and optimization perspectives, we identified a consistent depth-dependent adaptation pattern: alignment is not uniformly distributed but architecturally localized. Specifically, we characterize the top layers as sites of aggressive plasticity and information overwriting, whereas middle layers facilitate robust knowledge integration. Guided by these findings, our proposed Mid-Block Efficient Tuning exploits this stable intermediate zone, achieving substantial gains over full-depth approaches (e.g., +10.2% on GSM8K). Our findings suggest that future alignment strategies must move beyond uniform updates to prioritize the functional distinctiveness of model layers, balancing plasticity with stability to mitigate catastrophic forgetting.

Limitations

Our study focuses on establishing a mechanistic baseline for layer-wise alignment dynamics, however, we define the following scopes to ensure the precision of our claims, which point toward avenues for future generalization.

First, regarding architectural scope, we prioritized our analysis on standard dense decoder-only architectures (OLMo2 and Mistral families) to ensure a controlled experimental environment. While this enables us to isolate the effects of depth without the confounding routing noise of Mixture-of-Experts (MoE) or encoder-decoder interactions, extending our Mid-Block hypothesis to these complex topologies remains a promising direction for future research to verify cross-architecture universality.

Second, we explicitly target the SFT stage, isolating it from subsequent alignment phases like RLHF or DPO. This scoping is deliberate: it allows us to pinpoint the emergence of instruction-following capabilities at the representational level before preference optimization introduces reward-driven feature shifts. While our findings provide a solid foundation, investigating how these layer-wise dynamics evolve under preference-based objectives will be a valuable extension of this work.

Finally, while our proposed Mid-Block Efficient Tuning demonstrates significant gains by updating the 20%-80% depth range, this block selection is currently empirical. Our results indicate that this sweet spot is a broad, robust plateau rather than a sharp, sensitive peak, suggesting high transferability. However, developing an adaptive, training-free metric to automatically identify these optimal boundaries for unseen architectures could further streamline the deployment of this method.

Acknowledgments

This work was partially supported by the Hefei College Talent Research Fund Project (No. 24RC20), the Scientific Research Project of the Anhui Provincial Education Department (No. 2025AHGXZK40379), and the Natural Science Research Project of the Anhui Educational Committee (No. 2024AH040209). We also thank the anonymous reviewers for their constructive comments and suggestions.

References

Khai Loong Aw, Syrielle Montariol, Badr AlKhamissi, Martin Schrimpf, and Antoine Bosselut. 2024. Instruction-tuning aligns LLMs to the human brain. In *First Conference on Language Modeling*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tai, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai,

Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25(1).

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems.

Hongzhe Du, Weikai Li, Min Cai, Karim Saraipour, Zimin Zhang, Himabindu Lakkaraju, Yizhou Sun, and Shichang Zhang. 2025. How post-training reshapes llms: A mechanistic view on knowledge, truthfulness, refusal, and confidence.

Sreyan Ghosh, Chandra Kiran Reddy Evuru, Sonal Kumar, Ramaneswaran S, Deepali Aneja, Zeyu Jin, Ramani Duraiswami, and Dinesh Manocha. 2024. A closer look at the limitations of instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 15559–15589. PMLR.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326, Dublin, Ireland. Association for Computational Linguistics.

Zirui He, Haiyan Zhao, Yiran Qiao, Fan Yang, Ali Payani, Jing Ma, and Mengnan Du. 2025. Saif: A sparse autoencoder framework for interpreting and steering instruction following of language models.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Saffron Huang, Divya Siddarth, Liane Lovitt, Thomas I. Liao, Esin Durmus, Alex Tamkin, and Deep Ganguli. 2024. Collective constitutional ai: Aligning a language model with public input. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24, page 1395–1417, New York, NY, USA. Association for Computing Machinery.

Itay Itzhak, Gabriel Stanovsky, Nir Rosenfeld, and Yonatan Belinkov. 2024. Instructed to bias: Instruction-tuned language models exhibit emergent cognitive bias. *Transactions of the Association for Computational Linguistics*, 12:771–785.

Aditi Jha, Sam Havens, Jeremy Dohmann, Alex Trott, and Jacob Portes. 2023. Limit: Less is more for instruction tuning across evaluation paradigms.

Gangwei Jiang, Caigao JIANG, Zhaoyi Li, Siqiao Xue, JUN ZHOU, Linqi Song, Defu Lian, and Ying Wei. 2025. Unlocking the power of function vectors for characterizing and mitigating catastrophic forgetting in continual instruction tuning. In *The Thirteenth International Conference on Learning Representations*.

Po-Nien Kung and Nanyun Peng. 2023. Do models really learn to follow instructions? an empirical study of instruction tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1317–1328, Toronto, Canada. Association for Computational Linguistics.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2024. The unlocking spell on base LLMs: Rethinking alignment via in-context learning. In *The Twelfth International Conference on Learning Representations*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc.

Kaifeng Lyu, Haoyu Zhao, Xinran Gu, Dingli Yu, Anirudh Goyal, and Sanjeev Arora. 2024. Keeping llms aligned after fine-tuning: The crucial role of prompt templates. *Advances in Neural Information Processing Systems*, 37:118603–118631.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.

Mengjie Ren, Boxi Cao, Hongyu Lin, Cao Liu, Xianpei Han, Ke Zeng, Wan Guanglu, Xunliang Cai, and Le Sun. 2024. Learning or self-aligning? rethinking instruction fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6090–6105, Bangkok, Thailand. Association for Computational Linguistics.

Yi Ren and Danica J. Sutherland. 2025. Learning dynamics of LLM finetuning. In *The Thirteenth International Conference on Learning Representations*.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. 2024. Mathscale: scaling instruction tuning for mathematical reasoning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. *arXiv preprint arXiv:1905.05950*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Xuansheng Wu, Wenlin Yao, Jianshu Chen, Xiaoman Pan, Xiaoyang Wang, Ninghao Liu, and Dong Yu. 2024. From language modeling to instruction following: Understanding the behavior shift in LLMs after instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2341–2369, Mexico City, Mexico. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023a. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023b. Instruction-following evaluation for large language models.

A Sensitivity Analysis of Mid-Block Boundary Choices

To examine whether the performance of Mid-Block Efficient Tuning is sensitive to the choice of segment granularity, we extend our ablation studies by partitioning model layers into $M = 3$ and $M = 10$ equal segments, in addition to the $M = 5$ configuration reported in the main text. We evaluate on OLMo2-13B using the ToxiGen dataset, keeping all other experimental settings identical.

Table 1 reports results for $M = 3$ segments. The middle segment (Segment 2) achieves the highest accuracy (0.4721), outperforming both the first segment (0.4650) and the last segment (0.4550), confirming the inverted-U-shaped trend at a coarse granularity.

Table 1: Performance with $M = 3$ Segments (OLMo2-13B, ToxiGen).

Segment	Accuracy
Segment 1	0.4650
Segment 2	**0.4721**
Segment 3	0.4550

Table 2 reports results for $M = 10$ segments. Performance rises from Segment 1 (0.4503) to a peak at Segment 5 (0.4771), then declines sharply toward the final segments, with Segment 10 dropping to 0.3821. This finer-grained view reveals that the optimal zone corresponds precisely to the middle layers identified in our main analysis (approximately the 20%–80% depth range), and that performance degrades most severely in the tail segments closest to the output.

Table 2: Performance with $M = 10$ Segments (OLMo2-13B, ToxiGen).

Segment	Accuracy
Segment 1	0.4503
Segment 2	0.4589
Segment 3	0.4693
Segment 4	0.4761
Segment 5	**0.4771**
Segment 6	0.4754
Segment 7	0.4502
Segment 8	0.4215
Segment 9	0.3966
Segment 10	0.3821

Taken together, these results demonstrate that the inverted-U-shaped pattern is robust across segment granularities. As the number of segments increases, the pattern becomes more pronounced, with performance peaking in the exact middle layers and sharply degrading toward the output layers. This confirms that the 20%–80% interval is not a sensitive heuristic but rather a broad, stable region that is tolerant to minor boundary perturbations.

Table 3: Summary of instruction-tuned versions of commonly used open-source LLMs.

Model Family	Official Aligned Version	Alignment Pipeline	Suitable
LLaMA 2 / 3	LLaMA-2-Chat / LLaMA-3-Instruct	SFT + RLHF / DPO	✗
Gemma 1 / 2	Gemma-IT / Gemma-2-IT	SFT + RLHF	✗
Qwen 1.5 / 2	Qwen-Chat / Qwen2-Instruct	SFT + DPO	✗
DeepSeek	DeepSeek-LLM-Chat / V2	SFT + RLHF / DPO	✗
Falcon	Falcon-Instruct	Mixed / Unpaired	✗
Pythia	None (Base only)	N/A	✗
OPT	None (Base only / OPT-IML)	N/A	✗
OLMo2	OLMo-2-Instruct	Pure SFT	✓
Mistral	Mistral-7B-Instruct-v0.1	Pure SFT	✓

Table 4: Perplexity (↓) for pre-training knowledge retention across tuning strategies (OLMo2-13B).

Strategy	IFEval ↓	MT-Bench ↓
Base (`00000`)	14.53	13.58
Full-layer LoRA (`11111`)	14.70	13.76
Top-layer Segments (`10000`)	14.61	13.63
Tail-layer Segments (`00001`)	14.65	13.72
Middle-layer Segments (`00110`)	**14.54**	**13.60**

B Pre-training Knowledge Retention Analysis

A key motivation for Mid-Block Efficient Tuning is its potential to mitigate catastrophic forgetting by avoiding aggressive updates to the highly plastic final layers. To provide direct quantitative evidence for this claim, we measure the retention of pre-trained capabilities across different tuning strategies using perplexity on IFEval and MT-Bench, where lower perplexity indicates better preservation of pre-trained knowledge.

We compare five configurations on OLMo2-13B: the untuned Base model (`00000`), standard full-layer LoRA (`11111`), Top-layer Segments (`10000`), Tail-layer Segments (`00001`), and Middle-layer Segments (`00110`), following the same segment notation as in Section 4.4.

As shown in Table 4, Middle-layer Segments achieves perplexity closest to the Base model on both benchmarks (14.54 on IFEval and 13.60 on MT-Bench), substantially outperforming standard full-layer LoRA (14.70 and 13.76) and the tail-layer strategy (14.65 and 13.72). These results demonstrate that concentrating updates on intermediate layers effectively mitigates the catastrophic forgetting induced by standard full-layer LoRA, successfully preserving pre-trained capabilities while maintaining robust task alignment.

C Survey of Open-Source LLMs and Alignment Pipelines

A natural concern is whether our experimental conclusions generalize beyond the two model families studied (OLMo2 and Mistral-7B). We carefully selected these families after a thorough survey of other widely-used open-source LLMs, including LLaMA, Gemma, Qwen, DeepSeek, Falcon, Pythia, and OPT. Our investigation reveals that most of these models either (1) do not provide a corresponding instruction-tuned version trained exclusively with SFT, or (2) their instruction-tuned versions involve a combination of multiple alignment techniques such as SFT, RLHF, and DPO, which introduces confounding factors inconsistent with our experimental setup.

Table 3 summarizes the alignment pipelines of commonly used open-source LLMs. OLMo2 is selected for its full model family coverage across multiple scales (1B, 7B, 13B, 32B), and Mistral-7B is selected for its clean SFT-only instruction-tuned variant. Both are the only families that satisfy our requirement of a clean base, SFT model pair trained without additional preference optimization stages.

This survey confirms that OLMo2 and Mistral-7B represent the most appropriate and controlled experimental setting for isolating the effects of supervised fine-tuning, and that the limited architectural scope is a deliberate methodological choice rather than an oversight.