TRANSFORMER BLOCK COUPLING AND ITS CORRELATION WITH GENERALIZATION IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) have made significant strides in natural language processing, and a precise understanding of the internal mechanisms driving their success is essential. In this work, we trace the trajectories of individual tokens as they pass through transformer blocks, and linearize the system along these trajectories through their Jacobian matrices. By examining the relationships between these Jacobians, we uncover the **transformer block coupling** phenomenon in a multitude of LLMs, characterized by the coupling of their top singular vectors across tokens and depth. Our findings reveal that coupling *positively correlates* with model performance, and that this relationship is stronger than with other hyperparameters, namely parameter budget, model depth, and embedding dimension. We further investigate the emergence of these properties through training, noting the development of coupling, as well as an increase in linearity and layer-wise exponential growth in the token trajectories. These collective insights provide a novel perspective on the interactions between token embeddings, and prompt further approaches to study training and generalization in LLMs.

1 INTRODUCTION

In recent years, many openly available Large Language Models (LLMs) have been released, achieving state-of-the-art results on task-specific benchmarks. The abundance of models, each with differing architecture and training methodology, motivates comparing the underlying mechanisms that drive generalization.

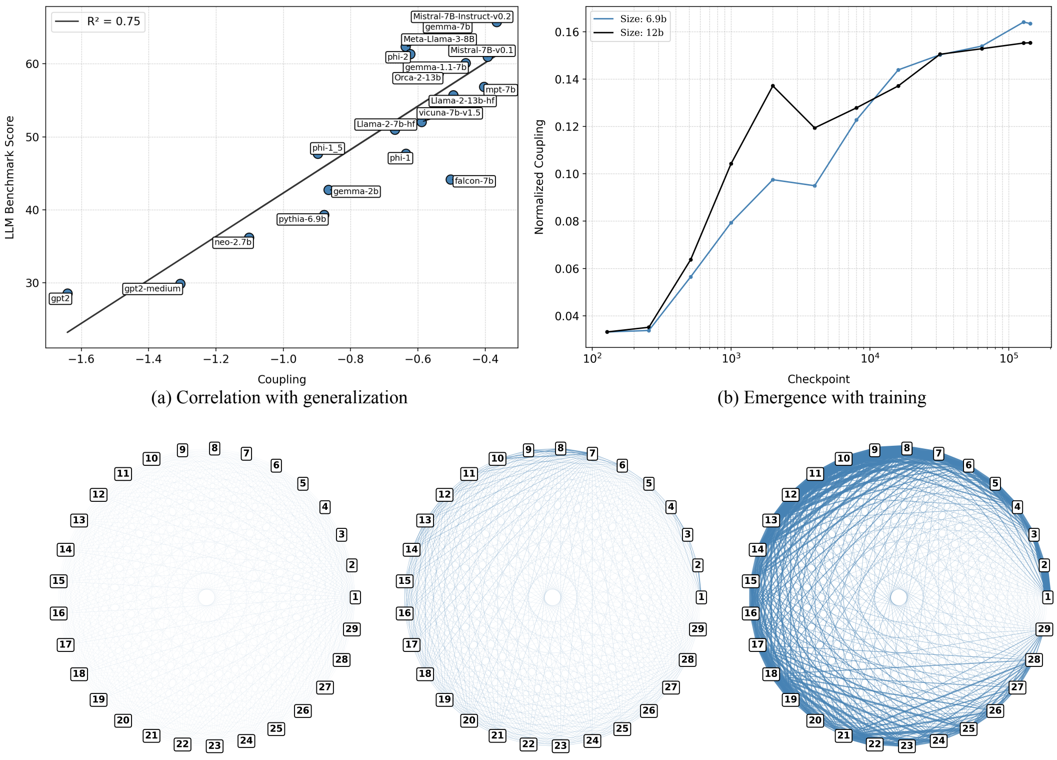
Transformers (Vaswani et al., 2017) can be represented as discrete, nonlinear, coupled dynamical systems, operating in high dimensions (Greff et al., 2016; Papyan et al., 2017; Haber & Ruthotto, 2017; Ee, 2017; Ebski et al., 2018; Chen et al., 2018; Bai et al., 2019; Rothauge et al., 2019; Gai & Zhang, 2021; Li & Papyan, 2023). Viewing the skip connections as enabling a discrete time step, we represent the hidden representations as dynamically evolving through the layers of the network. The term *nonlinear* refers to the nonlinear transformations introduced by activation functions, and *coupled* refers to the interdependent token trajectories that interact through the MLP and self-attention blocks.

In our work, we investigate whether there are identifiable structural characteristics across 38+ pretrained LLMs, measure their emergence with training, and analyze their relationship with generalization performance. During inference, as token embeddings pass through the network, we linearize the effect of transformer blocks on the token embeddings throughout the depth of the LLM. To this end, we compute the Jacobians of distinct connections between layers or tokens, derive their singular value decompositions (SVDs), and compare the resulting singular vectors. This approach measures the degree of coupling between singular vectors to capture the operational similarity of blocks as they act on tokens. This perspective raises several questions:

Q1. What regularity properties do these trajectories exhibit, and what are their relations with one another? More concretely, what is the relation between the Jacobians across different tokens and transformer layers?

Q2. How do the properties of hidden representations and their relations emerge with training?

Q3. Are any of these properties related to the generalization capabilities of LLMs?



(a) Correlation with generalization

(b) Emergence with training

(c) Pythia 12B **Checkpoint 1**

(d) Pythia 12B **Checkpoint 4000**

(e) Pythia 12B **Checkpoint 143000**

Figure 1: **Transformer Coupling Measurements**. **(a)** The plot illustrates the correlation between average coupling (negative mis-coupling) and benchmark scores across base LLMs (not fine-tuned), showing that higher alignment corresponds to improved performance, with a regression fit yielding an R^2 value of 0.75 with a significant p-value of 1.56×10^{-6}. **(b)** The mean normalized coupling (Section 3.1) is plotted as a function of training checkpoints for Pythia 12B and 6.9B (Biderman et al., 2023), measured at steps $128, 256, 512, 1k, 2k, \ldots, 128k, 143k$. **(c-e)** Adjacency plots illustrate the mean coupling scores between pairs of layers. Each node represents a layer, and edge weight and opacity indicate the strength of depth-wise normalized coupling. Visualizations are provided for checkpoints 1, 4k, and 143k of Pythia 12B.

1.1 Contributions

We investigate the motivating questions across several openly-available LLMs, most having over 1B parameters, trained by 6 independent organizations, with varying training methods and data (Appendix A.1). Through our experiments, we identify consistent patterns that define the **transformer block coupling** phenomenon.

1. **Coupling.** The singular vectors of the Jacobians of transformer blocks couple across depth (Figures 3, 20) and tokens (Figures 4, 18, 19, 21, 22, 23) in several open source LLMs (Table (1)). Further, coupling across Jacobians emerges with training (Figures 1b, 3, 4, 18, 19), and the coupling strength becomes more pronounced between adjacent layers with training, indicating a layer-wise locality in the interactions (Figures 1(c-e)).
2. **Generalization.** The strength of coupling is **correlated with** benchmark performance on the HuggingFace Open LLM Leaderboard (Beeching et al., 2023) (Figures 1a, 36). Additionally, coupling is more strongly correlated with generalization than parameter budget, model depth, and token embedding dimension (Figure 35).
3. **Regularity.** Linearity in hidden trajectories emerges with training (Figures 28, 12, 5(a), 14, 26, 27), aligning with behaviour previously observed in ResNets Li & Papyan (2023). Exponential growth occurs in contiguous token representations as a function of depth (Figures 5(b), 25, 26, 29), starkly contrasting linear growth previously observed.

We provide a new perspective on token embedding interactions within LLMs by examining layers of a transformer through their Jacobian matrices. Our results display the effect of training on transformer blocks, and suggest potential approaches for promoting generalization in LLMs.

2 Background on Large Language Models

We describe LLMs as a deep composition of functions that iteratively transform token embeddings. In the input layer, $l = 0$, textual prompts undergo tokenization and are combined with the positional encodings to create an initial high-dimensional embedding, denoted by $x_i^0 \in \mathbb{R}^{d_{\text{model}}}$ for the i^{th} token. When these embeddings are stacked, they form a matrix:

$$X^0 = (x_1^0, x_2^0, \ldots, x_n^0) \in \mathbb{R}^{n \times d_{\text{model}}}. \tag{1}$$

The embeddings then pass through L transformer blocks:

$$X^0 \xrightarrow{F_{\text{block}}^1} X^1 \xrightarrow{F_{\text{block}}^2} \cdots X^{L-1} \xrightarrow{F_{\text{block}}^L} X^L. \tag{2}$$

$X^l = F_{\text{block}}^l(X^{l-1})$ denotes the embeddings after the l^{th} block, consisting of causal multi-headed attention (MHA), a feed-forward network (FFN), and normalization layers (LN) with residual connections:

$$h^{l+1}(X^l) = \text{MHA}(\text{LN}(X^l)) \tag{3}$$
$$g^{l+1}(X^l) = \text{LN}(X^l + h^{l+1}(X^l)) \tag{4}$$
$$f^{l+1}(X^l) = h^{l+1}(X^l) + \text{FFN}(g^{l+1}(X^l)) \tag{5}$$
$$F_{\text{block}}^{l+1}(X^l) = X^l + f^{l+1}(X^l), \tag{6}$$

where the MHA, LN, FFN are implicitly indexed by layer. Among many models (Appendix 1), an additional rotary positional embedding (RoPE, Su et al. (2023)) is applied in the MHA layer. In the final representation, typically an additional layer normalization is applied:

$$F_{\text{block}}^L(X^{L-1}) = \text{LN}(X^{L-1} + h^L(X^{L-1}) + \text{FFN}(g^L(X^{L-1}))). \tag{7}$$

The output X^L from the final block F^L is passed into a bias-free linear layer $M \in \mathbb{R}^{d_{\text{vocab}} \times d_{\text{model}}}$, with d_{vocab} denoting the size of the token vocabulary and d_{model} is the dimension of the token embeddings. This layer M computes final-layer logits for each token embedding, $\ell_i = M x_i^L$. The prediction for the next token is then determined by selecting the maximal logit value: $\arg\max_{v \in \text{tokens}} \ell_{v,n}$.

3 Methods

3.1 Coupling of Singular Vectors of Jacobians

Jacobians. Coupling is investigated through analyzing the linearizations of transformer blocks which is given by their Jacobian matrices

$$J_{t_1 t_2}^l = \frac{\partial}{\partial x_{t_1}^{l-1}} \left(f^l(X^{l-1}) \right)_{t_2}, \tag{8}$$

defined for each layer $l \in \{1, \ldots, L\}$, and pair of tokens $t_1, t_2 \in \{1, \ldots, n\}$. Note that this is the Jacobian matrix for each transformer block without the contribution from the skip connection from the input of the block, similar to the quantity measured by Li & Papyan (2023) which strictly analyzes the case where $t_1 = t_2$.

Due to the causal structure of the representations, $J_{t_1 t_2}^l = 0$ whenever $t_1 > t_2$. Hence, we restrict our attention to the case where $t_1 \leq t_2$.

Singular value decomposition (SVD). We compute the SVD of the Jacobians:

$$J_{t_1 t_2}^l = U_l S_l V_l^\top$$

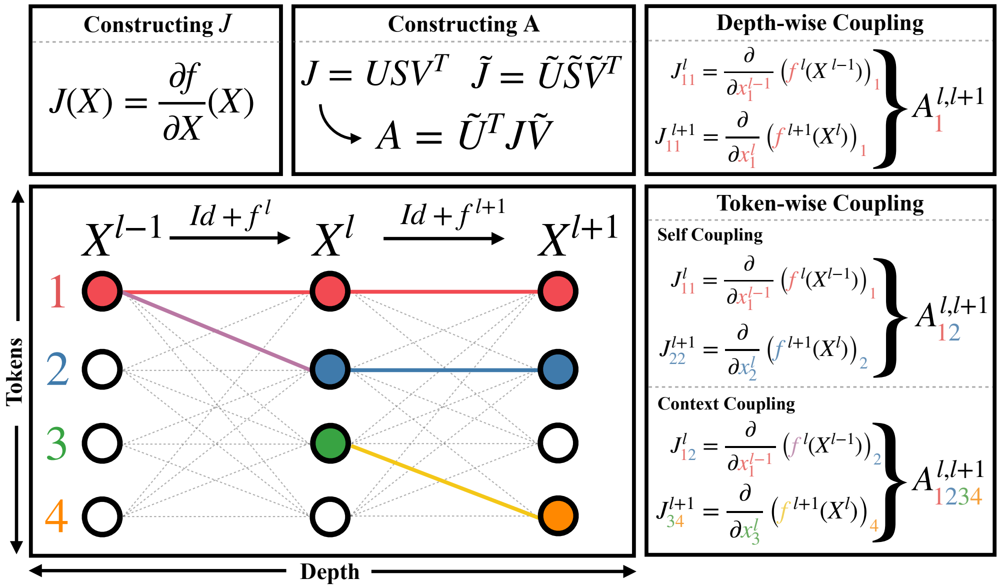


Figure 2: **Transformer Block Coupling.** A visualization of the various types of transformer block coupling with brief instructions on computing both the Jacobians J and coupling matrices A (Section 3.1). The coupling measurement quantifies the alignment and agreement between the interactions of embeddings connections within the network. The colored subscripts in the sample matrices A indicate the specific connections being compared.

where $U_l, V_l \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ are the matrices of left and right singular vectors respectively, and $S_l \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ contains the singular values.[1]

Coupling Measurement. To measure coupling between two Jacobians

$$J^l_{t_1 t_2} = U_l S_l V_l^\top \qquad \text{and} \qquad J^{l'}_{t'_1 t'_2} = U_{l'} S_{l'} V_{l'}^\top, \tag{9}$$

we define the *coupling matrix*

$$A^{ll'}_{t_1 t_2 t'_1 t'_2} := U_{l'}^\top J^l_{t_1 t_2} V_{l'} \tag{10}$$

$$= U_{l'}^\top U_l S_l V_l^\top V_{l'} \tag{11}$$

for $l, l' \in \{1, \ldots, L\}$ and $t_1, t_2, t'_1, t'_2 \in \{1, \ldots, n\}$. If the singular vectors of distinct Jacobians are strongly aligned, then

$$U_{l'}^\top U_l \approx I \approx V_l^\top V_{l'}, \tag{12}$$

implying that the coupling matrix A should be strongly diagonal. Explicitly, we quantify the mis-coupling of A using

$$m(A) = \|A - \text{Diag}(A)\|_F, \tag{13}$$

where $\text{Diag}(A)$ is the matrix A with all non-diagonal entries replaced by zero and $\|\cdot\|_F$ denotes the Frobenius norm. For normalized comparison between models, we normalize by A;

$$\tilde{m}(A) = \frac{\|A - \text{Diag}(A)\|_F}{\|A\|_F}. \tag{14}$$

Depth-wise Coupling. To analyze coupling across transformer blocks, we fix t, and measure alignment between J^l_{tt} and $J^{l'}_{tt}$ through the matrix $A^{ll'}_t$ across layers $l, l' \in \{1, \ldots, L\}$ [2]

Token-wise Coupling We also quantify the coupling across tokens in several ways:

[1]Note that the superscripts t_1, t_2, indicating the tokens, are omitted for clarity in the expression for the SVD.
[2]In the matrix A, we write the single subscript t for clarity.

- **Self-coupling.** By fixing two layers $l, l' \in \{1, \ldots, L\}$, we analyze the case where the input and output tokens are the same. Explicity, we compare J^l_{tt} and $J^{l'}_{t't'}$ across $t, t' \in \{1, \ldots, n\}$, which represents the coupling across tokens for a token's effect on its own trajectory.
- **Context Coupling.** We consider the context tokens' impact on a trajectory by measuring coupling between $J^l_{t_1t_2}$ and $J^{l'}_{t_1t'_2}$ across $t_2, t'_2 \geq t_1$ (fixing the input token to be the same) and also between $J^l_{t_1t_2}$ and $J^{l'}_{t'_1t_2}$ across $t_1, t'_1 \leq t_2$ (fixing the output token to be the same).

3.2 Linearity of Trajectories

Linearity in intermediate embeddings is quantified with the *line-shape score* (LSS), defined by Gai & Zhang (2021) as

$$\text{LSS}_i^{0,\ldots,L} = \frac{L}{\left|\left|\tilde{x}_i^L - \tilde{x}_i^0\right|\right|_2}, \tag{15}$$

where $\tilde{x}_i^0 = x_i^0$, i.e., the input embeddings passed to the LLM, and $\tilde{x}_i^l$ is defined recursively as

$$\tilde{x}_i^l = \tilde{x}_i^{l-1} + \frac{x_i^l - x_i^{l-1}}{\left|\left|x_i^l - x_i^{l-1}\right|\right|_2} \text{ for } l = 1, \ldots, L. \tag{16}$$

Note that $\text{LSS} \geq 1$, with $\text{LSS} = 1$ if and only if the intermediate representations $x_i^0, \ldots, x_i^L$ form a co-linear trajectory.

3.3 Layer-Wise Exponential Growth

We measure the presence of exponential spacing (*expodistance*) of the hidden trajectories. Assuming exponential growth of the embedding norms as they flow through the hidden layers, we estimate $\|x_i^l\| \approx e^{\alpha l}\|x_i^0\| = e^{\alpha}\|x_i^{l-1}\|$ for some fixed $\alpha \in \mathbb{R}$ over all layers $l = 1, \ldots L$. We quantify the validity of this representation by measuring the coefficient of variation of α_i^l, given by

$$\alpha_i^l \approx \ln\left(\frac{\|x_i^l\|}{\|x_i^{l-1}\|}\right), \tag{17}$$

for each layer l and token i. Under exponential growth, it is expected that α_i^l is independent of depth. We therefore denote the expodistance (ED) of the trajectory of the i^{th} token of a given sequence by

$$\text{ED}_i = \frac{\text{Var}_l \alpha_i^l}{(\text{Avg}_l \alpha_i^l)^2}. \tag{18}$$

This measurement is motivated by the discussion in Section 6.1 the parametrization discussed in Appendix A.5, as well as empirical evidence in Figure 30a, and serves as a method to test the validity of the linearization presented in Equation 19.

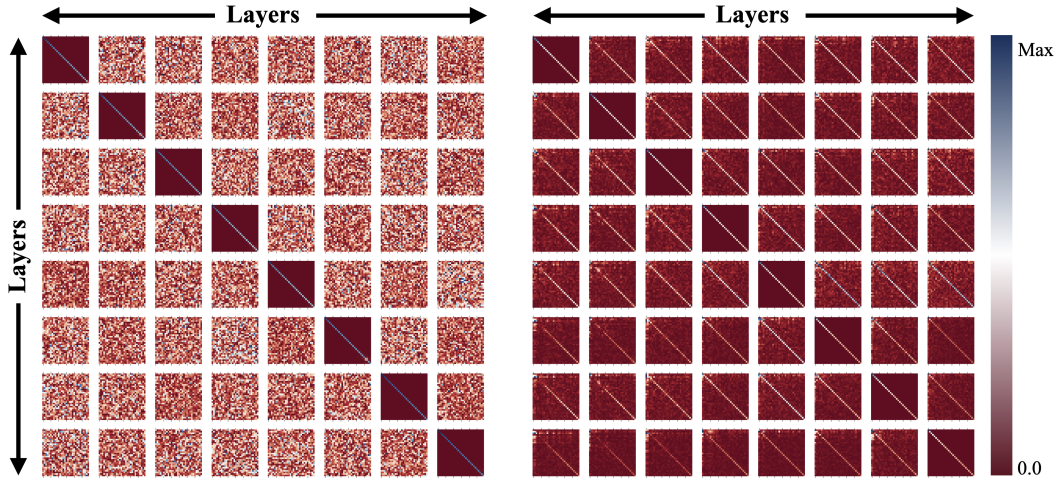
3.4 Vision Transformer Training

For further investigation of coupling in transformers, we train a series of Vision Transformers (ViTs) following DEiT training (Touvron et al., 2021). We train 64 ViTs on CIFAR10 (Krizhevsky, 2009) with varied weight decay and stochastic depth rate for a fixed architecture of embedding size 192, depth 12, and 3 attention heads. Please see Appendix A.7 for further details.

4 Evaluation

4.1 Suite of Large Language Models

Our study evaluates a total of 38 LLMs (24 base LLMs and 14 fine-tuned, see Appendix A.1) that were independently trained by various individuals and organizations. These models, provided



(a) Llama-3 8B **Untrained**

(b) Llama-3 8B **Trained**

Figure 3: **Transformer Block Coupling across Depth**. The figure shows Jacobian coupling across transformer blocks 9 to 16, using the prompt "What is the capital of France? The capital is" to trace the final token's trajectory. In trained models (bottom row), the diagonal pattern with minimal off-diagonal values indicates alignment of Jacobians, where top singular vectors of $J^{l'}$ diagonalize J^{l}. Untrained models (top row) lack this alignment. Further details are in the Appendix. A.8 (Figure 20). Best viewed in color.

through HuggingFace (Wolf et al., 2020), vary in terms of parameter budgets, number of layers, hidden dimensions, and training tokens. Moreover, we analyze the dynamics of each measurement throughout training by deploying the Pythia Scaling Suite (Biderman et al., 2023). A summary of the models under consideration is presented in Table 1 of Appendix A.1 and further details in Appendix A.6.

4.2 Prompt Data

We evaluate these LLMs using prompts of varying length, ambiguity, and context, sourced from the test set of ARC (Clark et al., 2018), GSM8K (Cobbe et al., 2021), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), Truthful QA (Lin et al., 2022), and Winogrande (Sakaguchi et al., 2019). This data sets the performance benchmarks on the HuggingFace Open LLM Leaderboard (Beeching et al., 2023) and provide a representative evaluation of performance on many language tasks.

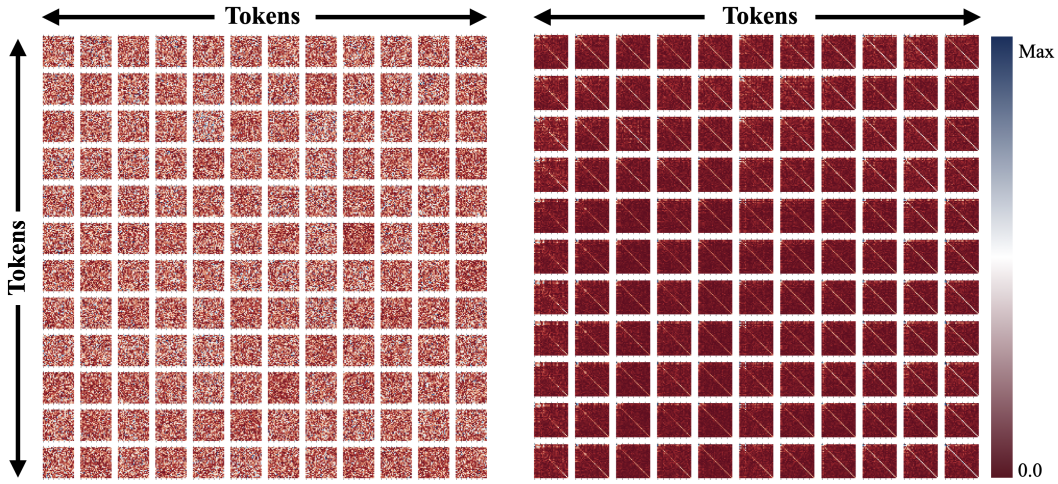
5 Results

5.1 Coupling of Jacobians Across Depth

In trained LLMs, we observe coupling of the top singular vectors of the Jacobians across depth (Figure 3 bottom row), evident in the low non-diagonal values with a visible diagonal present in the matrix subplots. This is consistently observed across various LLMs considered. On the other hand, in untrained models (Figure 3 top row), there is no coupling of Jacobians across different depths. There is coupling along the diagonal, however, because each Jacobian is trivially diagonalized by its own singular vectors. This, in addition to Figure 1, suggests that coupling across depth emerges through training.

5.2 Coupling of Jacobians Across Tokens

We analyze the coupling of singular vectors of Jacobians across tokens. For input and output tokens that are the same (J_l^{tt} and $J_{l'}^{t't'}$, Figure 4), we observe strong coupling, indicating that a token's interactions along its trajectory are coupled with others. For context tokens, coupling is examined by fixing the input token ($J_l^{t_1 t_2}$ and $J_{l'}^{t_1 t_2'}$, Figure 18) or the output token ($J_l^{t_1 t_2}$ and $J_{l'}^{t_1' t_2}$, Figure 19). While coupling exists, it is less consistent across pairs. Untrained models (Figure 4 top row) show no such coupling.



(a) Llama-3 8B **Untrained**

(b) Llama-3 8B **Trained**

Figure 4: **Transformer Block Coupling across Tokens (Same input and output tokens)**. The figure shows Jacobian coupling for the same input and output token across tokens, visualized using the absolute values of $A_{ll'}^{ttt't'}$ (with fixed layers l, l'). In trained models (bottom row), the strong diagonal and small off-diagonal values indicate coupling, while no such coupling is present at initialization (top row). Additional details are in Appendix A.8 (Figure 21).

5.3 Emergence of Coupling with Training

Coupling emerges through training for the evaluated LLMs, including coupling across depth (Figure 3) and across tokens (Figures 4, 18, 19). Further, we evaluate layer-wise coupling at intermediate training checkpoints of Pythia 6.9B and 12B (Biderman et al., 2023) (Figure 1b), and observe that coupling is generally low at initialization and increases persistently throughout training. Moreover, there is a clear sense of locality in the strength of coupling which is visually displayed in Figure (1c-e)

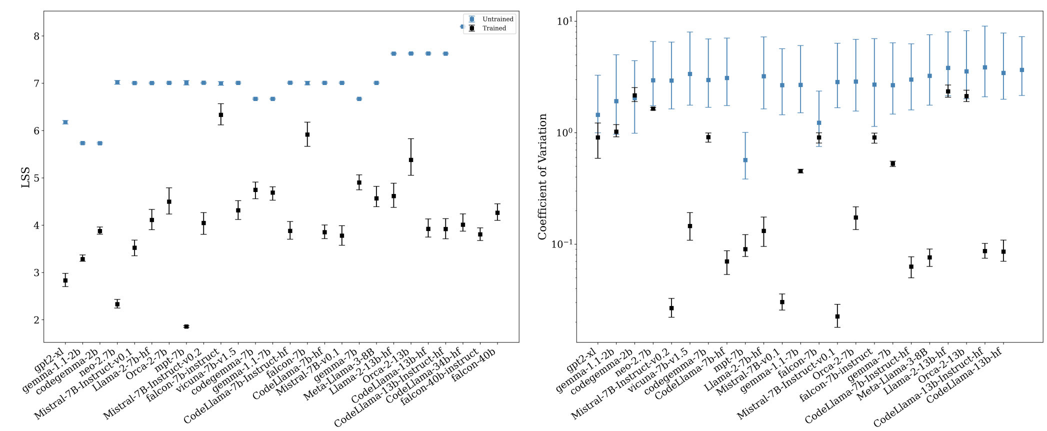
The gradual growth of coupling observed in Figure (1b) parallels the logarithmic increase in accuracy during training for Pythia 12B and Pythia 6.9B (Biderman et al., 2023). This highlights the relationship between coupling and performance, since both properties emerge at similar training iterations and rates.

5.4 Correlation with Generalization

For each LLM, we measure the average coupling across depth(which we define to be negative mis-coupling) across prompts in the 6 evaluation datasets (Section 4.2), where for each prompt, $m(A^n_{ll',K})$ is averaged over layers $l, l' \in \{1, \dots, L\}$. We plot the coupling values against the benchmark scores across several LLMs (Figure 1). Our results reveal a positive correlation between coupling and performance benchmark scores, and is more significant than the relationship between other significant model hyperparameters (Figure 35). This observation suggests a compelling relationship between stronger coupling of singular vectors of Jacobians $J_l^{t_1 t_2}$ and improved generalization.

The results for ViTs demonstrate that stochastic depth encourages coupling during training (Figure 6b) and that coupling correlates with accuracy when fixing SD rate (Figure 6a). This finding suggests that coupling may provide new insight into stochastic depth's underlying mechanism, and that developing training methods to amplify coupling across transformer blocks could provide additional regularization and improve performance.

We hypothesize that simple trajectories may lead to better generalization. This agrees with many generalization bounds in machine learning (Arora et al., 2018), which suggest that models with lower complexity tend towards better generalization. Additionally, prior works (Novak et al., 2018) demonstrate that the Frobenius norm of input-output Jacobians is related to generalization, providing evidence that coupling — a structural property derived from Jacobians — may also correlate with generalization.



(a) Linearity (b) Expodistance

Figure 5: **Regularity of Trajectories.** The figure depicts the line-shape score (LSS) of embedding trajectories, as discussed in Section 5.5, computed on 1,200 prompts of the HuggingFace Open LLM Leaderboard (Section 4.2) for a variety of trained (black) and randomly initialized (blue) LLMs (Appendix A.1). Plotted are the median values over all prompts, and are accompanied with uncertainty intervals depicting the inter-quartile range of the results for each model. Models are sorted by number of parameters.

5.5 Regularity in Hidden Trajectories

We identify a considerable degree of linearity in the hidden trajectories of all featured LLMs. The LSS of the trajectories has an average value 4.25 in trained models, while taking average values of 6.54 at initialization, and is supported by the low variation across benchmark prompts (Figure 5). Linearity increases with training at varying depths of Pythia 12B (Figure 28). Linear and expansive behavior of the representations is demonstrated in Llama-3 70B, MPT 30B, and NeoX 20B through low dimensional projections of embedding trajectories (Figure 12).

Among the LLMs considered, the vast majority of hidden trajectories exhibit exponential growth that emerges with training (Figure 5(b) b). Many models exhibit a low coefficient of variation across prompts, showing the robustness of this property across a variety of tasks. In contrast, measurements at initialization show equally (rather than exponentially) distanced trajectories, as reflected by the low coefficients of variation in the norms of their layer-wise differences (Figure 25). Under certain assumptions, exponential spacing is motivated in Section 6.1.

6 Discussion

6.1 Emergence of Regularity with Coupling

Under certain assumptions, the emergence of increased linearity and exponential spacing in many LLMs can be analyzed as a result of coupling. Considering input embeddings $x_1^0, \ldots, x_n^0$ and the linearization of the last token embedding x_n^l given by $J_{n,n}^l(x_1^0, \ldots, x_n^0)$:

$$x_n^l = (I + J_{n,n}^l(x_1^0, \ldots, x_n^0))x_n^{l-1}$$

We simplify notation and write $x_n^l = x^l$, $J_{n,n}^l(x_1^0, \ldots, x_n^0) = J^l$. The representations follow the linearized equation:

$$x^{l+1} = x^l + J_l x^l = (I + J_l)x^l.$$

Expanding across layers, the entire system can be approximated by the product

$$x^L = (I + J_L)(I + J_{L-1}) \cdots (I + J_1)x^0.$$

Assuming that $J_l \approx USU^T$, this equation predicts that the norm of x_l would exhibit exponential growth layer by layer. Expanding U and S,

$$x^l = \sum_{j=1}^{d_{\text{model}}} u_j (1 + s_j)^l u_j^\top x^0.$$

where u_j and s_j represent the eigenvectors and eigenvalues of the Jacobian, respectively. In general, trajectories are not expected to be perfectly linear unless x^0 aligns with an eigenvector of J. However, in our experiments, we observe a notable tendency towards linearity, suggesting that the representations align progressively during training with the eigenvectors of the coupled Jacobians.

6.2 Significance of Coupling

The coupling phenomenon provides insight into the internal operations of transformer. We hypothesize that during training, the LLM learns to represent embeddings in specific low-dimensional subspaces (Eldar & Mishali, 2009). Given an input, the first layer converts the input into embeddings within one of these learned subspaces. Each subsequent transformer layer modifies these embeddings, potentially moving them to different subspaces. Strong coupling between consecutive layers suggests that the LLM tends towards representations in the same or similar subspaces across many layers. Weak coupling suggests that the subspaces may change between layers, though usually gradually, and that adjacent layers still operate in relatively similar subspaces. Previous works have shown (Lad et al., 2024; Gromov et al., 2024) that the early and late layers of language models behave differently, which may be understood through coupling; tokens remain in similar spans, then transition to a different subspace, continuing within a new span that is consistent in the remainder of the transformer.

The emergence of coupling with training steps (Figure 1) may provide insight into the dynamics. Under full coupling and a difference equation approximation, the representations evolve as

$$x^l = \sum_{j=1}^{d_{\text{model}}} u_j (1 + s_j)^l u_j^\top x^0$$

where u_i and λ_i denote the eigenvectors and eigenvalues of the Jacobian, respectively. In this case, the gradients of the loss L with respect to prediction y, x^L are represented by

$$\frac{\partial \mathrm{L}}{\partial x^0}(x^L, y) \approx \sum_{j=1}^{d_{\text{model}}} u_j (1 + s_j)^L u_j^\top (y - x^L),$$

Due to the coupling, the dynamics exhibit either exponential growth or decay in different subspaces, depending on the sign of s_j, which is known to cause challenges for optimization as in past works on dynamical isometry (Pennington et al., 2017). We infer that increasing coupling during training makes optimization progressively more difficult. Conversely, as training progresses, it becomes harder to achieve stronger coupling, and is consistent with the logarithmic trend in Figure 1.

7 Related Work

Residual Networks. ResNets (He et al., 2016) have been viewed as an ensemble of shallow networks (Veit et al., 2016), with studies delving into the scaling behaviour of their trained weights (Cohen et al., 2021). The linearization of residual blocks by their Residual Jacobians was first explored by Rothauge et al. (2019), who examined Residual Jacobians and their spectra in the context of stability analysis, and later by Li & Papyan (2023) who discovered Residual Alignment. Coupling of Jacobian singular vectors in LLM transformers extends previous results for classifier Resnets (Li & Papyan, 2023). We show coupling in $J_l^{t_1 t_2}$ across various tokens, which is specific to tokenization in transformers, in addition to demonstrating coupling of $J_l^{t_1 t_2}$ across l, which was also identified analogously in ResNets (Li & Papyan, 2023). Further comparison is included in Appendix A.3.

Neural Ordinary Differential Equations. Neural ODEs (Chen et al., 2018) view ResNets as a discretized dynamical process, with past work (Sander et al., 2022) showing the convergence of Residual Networks to a system of linear ODE, with some extensions to transformers (Zhong et al., 2022; Li et al., 2021) The emergence of coupling in transformers suggests that a discretization of a simple itertaive process emerges in LLMs.

In-context learning. LLMs can perform tasks through examples provided in a single prompt, demonstrating in-context learning (von Oswald et al., 2023; Bai et al., 2024; Ahn et al., 2023; Akyürek et al., 2023; Xie et al., 2021; Hahn & Goyal, 2023; Xing et al., 2024). Studies suggest trained self-attention layers implement gradient-descent-like updates across depth to minimize the MSE of a linear model:

$$x_{\text{final}} = \min_x ||Ax - b||^2.$$

These updates take the form:

$$x_{t+1} = (I + \epsilon A^T A)x_t - \epsilon A^T b.$$

Coupling across depth suggests similarity in the matrices $I + \epsilon A^T A$.

Hidden Representation Dynamics. Prior research interprets deep neural networks through dynamical systems, revealing that training trajectories align with geodesic curves (Gai & Zhang, 2021) and partition activation space into basins of attraction (Nam et al., 2023). For further related works, see (Geshkovski et al., 2023b;a; Tarzanagh et al., 2023; Valeriani et al., 2023).

Structure in Hidden Representations. Neural Collapse (Papyan et al., 2020) highlights emergent regularity in last-layer representations, with subsequent studies exploring hidden-layer structures and their theoretical underpinnings (Wang et al., 2024a; Parker et al., 2023; Zangrando et al., 2024; Garrod & Keating, 2024; Wang et al., 2024b; Hoyt & Owen, 2021; Arous et al., 2023; Zarka et al., 2021; Ben-Shaul & Dekel, 2022; Papyan, 2020; Súkeník et al., 2023). In LLMs, recent works identify uniform token structures (Wu & Papyan, 2024) and low-dimensional hidden trajectories (Sarfati et al., 2024). Our work examines local token interactions through Jacobian dynamics across all LLM layers.

8 Limitations

Our analysis is limited to pretrained LLMs and their fine-tuned variants due to the high computational cost of training large models. Variations in experimental setups across independently trained models hinder direct comparisons, making it challenging to pinpoint causes of differing regularity. However, the consistent emergence of these properties warrants further study.

9 Conclusion

Our primary goal was to contribute to the understanding of the mechanics underlying transformer architectures through an analysis of the trajectories of token embeddings and their interactions. Our research builds on the understanding of transformer architectures by revealing the coupling, across depth and token, of singular vectors in the Jacobians of transformer blocks for multiple LLMs trained by various organizations. We establish a correlation between the strength of this coupling and benchmark performance on the HuggingFace Open LLM Leaderboard, highlighting the significance of transformer block coupling for generalization. These findings open avenues for future research, encouraging deeper exploration into the connections between regularity of hidden representations, model specifications, and generalization.

10 Reproducibility Statement

Source code for reproducing measurements detailed in Section 3 is included as supplementary material. Additional implementation details for evaluation are included in Appendix A.6.

REFERENCES

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning, 2023.

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models, 2023.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. Falcon-40B: an open large language model with state-of-the-art performance. 2023.

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In *International conference on machine learning*, pp. 254–263. PMLR, 2018.

Gerard Ben Arous, Reza Gheissari, Jiaoyang Huang, and Aukosh Jagannath. High-dimensional sgd aligns with emerging outlier eigenspaces, 2023.

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. Slicegpt: Compress large language models by deleting rows and columns, 2024. URL https://arxiv.org/abs/2401.15024.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *Advances in Neural Information Processing Systems*, 32, 2019.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, 36, 2024.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. Open llm leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard, 2023.

Ido Ben-Shaul and Shai Dekel. Nearest class-center simplification through intermediate layers. In Alexander Cloninger, Timothy Doster, Tegan Emerson, Manohar Kaul, Ira Ktena, Henry Kvinge, Nina Miolane, Bastian Rieck, Sarah Tymochko, and Guy Wolf (eds.), *Proceedings of Topological, Algebraic, and Geometric Learning Workshops 2022*, volume 196 of *Proceedings of Machine Learning Research*, pp. 37–47. PMLR, 25 Feb–22 Jul 2022. URL https://proceedings.mlr.press/v196/ben-shaul22a.html.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling, 2023.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, March 2021. URL https://doi.org/10.5281/zenodo.5297715. If you use this software, please cite it using these metadata.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. Gpt-neox-20b: An open-source autoregressive language model, 2022.

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

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018.

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. Training verifiers to solve math word problems, 2021.

Alain-Sam Cohen, Rama Cont, Alain Rossier, and Renyuan Xu. Scaling properties of deep residual networks, 2021.

Mucong Ding, Chenghao Deng, Jocelyn Choo, Zichu Wu, Aakriti Agrawal, Avi Schwarzschild, Tianyi Zhou, Tom Goldstein, John Langford, Anima Anandkumar, and Furong Huang. Easy2hard-bench: Standardized difficulty labels for profiling llm performance and generalization, 2024. URL https://arxiv.org/abs/2409.18433.

Amir Ben Dror, Niv Zehngut, Avraham Raviv, Evgeny Artyomov, Ran Vitek, and Roy Jevnisek. Layer folding: Neural network depth reduction using activation linearization, 2021. URL https://arxiv.org/abs/2106.09309.

Stanisław Jastrz Ebski, Devansh Arpit, Nicolas Ballas, Vikas Verma, Tong Che, and Yoshua Bengio. Residual connections encourage iterative inference. In *International Conference on Learning Representations*, 2018.

Weinan Ee. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5:1–11, 02 2017. doi: 10.1007/s40304-017-0103-z.

Yonina C. Eldar and Moshe Mishali. Robust recovery of signals from a structured union of subspaces. *IEEE Transactions on Information Theory*, 55(11):5302–5316, 2009. doi: 10.1109/TIT.2009.2030471.

Sara Elkerdawy, Mostafa Elhoushi, Abhineet Singh, Hong Zhang, and Nilanjan Ray. To filter prune, or to layer prune, that is the question, 2020. URL https://arxiv.org/abs/2007.05667.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models, 2023. URL https://arxiv.org/abs/2305.10924.

Yonggan Fu, Haichuan Yang, Jiayi Yuan, Meng Li, Cheng Wan, Raghuraman Krishnamoorthi, Vikas Chandra, and Yingyan Lin. Depthshrinker: A new compression paradigm towards boosting real-hardware efficiency of compact neural networks, 2022. URL https://arxiv.org/abs/2206.00843.

Kuo Gai and Shihua Zhang. A mathematical principle of deep learning: Learn the geodesic curve in the wasserstein space, 2021.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Connall Garrod and Jonathan P. Keating. Unifying low dimensional observations in deep learning through the deep linear unconstrained feature model, 2024.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. The emergence of clusters in self-attention dynamics. 2023a.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. A mathematical perspective on transformers, 2023b.

Klaus Greff, Rupesh K Srivastava, and Jürgen Schmidhuber. Highway and residual networks learn unrolled iterative estimation. *arXiv preprint arXiv:1612.07771*, 2016.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. The unreasonable ineffectiveness of the deeper layers, 2024. URL https://arxiv.org/abs/2403.17887.

Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *CoRR*, abs/1705.03341, 2017. URL http://arxiv.org/abs/1705.03341.

Michael Hahn and Navin Goyal. A theory of emergent in-context learning as implicit structure induction. *arXiv preprint arXiv:2303.07971*, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021.

Christopher R. Hoyt and Art B. Owen. Probing neural networks with t-sne, class-specific projections and a guided tour, 2021.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL `https://arxiv.org/abs/2106.09685`.

Metod Jazbec, Patrick Forré, Stephan Mandt, Dan Zhang, and Eric Nalisnick. Early-exit neural networks with nested prediction sets, 2024. URL `https://arxiv.org/abs/2311.05931`.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

Jinuk Kim, Marwa El Halabi, Mingi Ji, and Hyun Oh Song. Layermerge: Neural network depth compression through layer pruning and merging, 2024. URL `https://arxiv.org/abs/2406.12837`.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023. URL `https://arxiv.org/abs/2205.11916`.

A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Toronto*, 2009.

Vedang Lad, Wes Gurnee, and Max Tegmark. The remarkable robustness of llms: Stages of inference?, 2024. URL `https://arxiv.org/abs/2406.19384`.

Bei Li, Quan Du, Tao Zhou, Shuhan Zhou, Xin Zeng, Tong Xiao, and Jingbo Zhu. Ode transformer: An ordinary differential equation-inspired model for neural machine translation, 2021.

Jianing Li and Vardan Papyan. Residual alignment: Uncovering the mechanisms of residual networks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Yuanzhi Li, Yingyu Liang, and Andrej Risteski. Recovery guarantee of weighted low-rank approximation via alternating minimization, 2016. URL `https://arxiv.org/abs/1602.02262`.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods, 2022.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers, 2021. URL `https://arxiv.org/abs/2106.04647`.

Andrew Nam, Eric Elmoznino, Nikolay Malkin, Chen Sun, Yoshua Bengio, and Guillaume Lajoie. Discrete, compositional, and symbolic representations through attractor dynamics, 2023.

Roman Novak, Yasaman Bahri, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study, 2018. URL `https://arxiv.org/abs/1802.08760`.

Vardan Papyan. Traces of class/cross-class structure pervade deep learning spectra, 2020.

Vardan Papyan, Yaniv Romano, and Michael Elad. Convolutional neural networks analyzed via convolutional sparse coding. *The Journal of Machine Learning Research*, 18(1):2887–2938, 2017.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020. doi: 10.1073/pnas.2015509117. URL `https://www.pnas.org/doi/abs/10.1073/pnas.2015509117`.

Liam Parker, Emre Onal, Anton Stengel, and Jake Intrater. Neural collapse in the intermediate hidden layers of classification neural networks, 2023.

Jeffrey Pennington, Samuel S. Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice, 2017. URL `https://arxiv.org/abs/1711.04735`.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Kai Rothauge, Zhewei Yao, Zixi Hu, and Michael W. Mahoney. Residual networks as nonlinear systems: Stability analysis using linearization, 2019.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code llama: Open foundation models for code, 2024.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WINOGRANDE: an adversarial winograd schema challenge at scale, 2019.

Michael E. Sander, Pierre Ablin, and Gabriel Peyré. Do residual neural networks discretize neural ordinary differential equations?, 2022.

Raphaël Sarfati, Toni J. B. Liu, Nicolas Boullé, and Christopher J. Earls. Lines of thought in large language models, 2024. URL `https://arxiv.org/abs/2410.01545`.

Simone Scardapane, Michele Scarpiniti, Enzo Baccarelli, and Aurelio Uncini. Why should we add early exits to neural networks? *Cognitive Computation*, 12(5):954–966, June 2020. ISSN 1866-9964. doi: 10.1007/s12559-020-09734-4. URL `http://dx.doi.org/10.1007/s12559-020-09734-4`.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL `https://arxiv.org/abs/2104.09864`.

Peter Súkeník, Marco Mondelli, and Christoph Lampert. Deep neural collapse is provably optimal for the deep unconstrained features model, 2023.

Davoud Ataee Tarzanagh, Yingcong Li, Xuechen Zhang, and Samet Oymak. Max-margin token selection in attention mechanism, 2023.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko

Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology, 2024.

MosaicML NLP Team. Introducing mpt-30b: Raising the bar for open-source foundation models, 2023a. URL www.mosaicml.com/blog/mpt-30b. Accessed: 2023-06-22.

MosaicML NLP Team. Introducing mpt-7b: A new standard for open-source, commercially usable llms, 2023b. URL www.mosaicml.com/blog/mpt-7b. Accessed: 2023-05-05.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers distillation through attention, 2021. URL https://arxiv.org/abs/2012.12877.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

Lucrezia Valeriani, Diego Doimo, Francesca Cuturello, Alessandro Laio, Alessio Ansuini, and Alberto Cazzaniga. The geometry of hidden representations of large transformer models, 2023.

Tycho F. A. van der Ouderaa, Markus Nagel, Mart van Baalen, Yuki M. Asano, and Tijmen Blankevoort. The llm surgeon, 2024. URL https://arxiv.org/abs/2312.17244.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Andreas Veit, Michael Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks, 2016.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent, 2023.

Peng Wang, Xiao Li, Can Yaras, Zhihui Zhu, Laura Balzano, Wei Hu, and Qing Qu. Understanding deep representation learning via layerwise feature compression and discrimination, 2024a.

Sicong Wang, Kuo Gai, and Shihua Zhang. Progressive feedforward collapse of resnet training, 2024b.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020.

Robert Wu and Vardan Papyan. Linguistic collapse: Neural collapse in (large) language models, 2024. URL https://arxiv.org/abs/2405.17767.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Yue Xing, Xiaofeng Lin, Namjoon Suh, Qifan Song, and Guang Cheng. Benefits of transformer: In-context learning in linear regression tasks with unstructured data. *arXiv preprint arXiv:2402.00743*, 2024.

Emanuele Zangrando, Piero Deidda, Simone Brugiapaglia, Nicola Guglielmi, and Francesco Tudisco. Neural rank collapse: Weight decay and small within-class variability yield low-rank bias, 2024.

John Zarka, Florentin Guth, and Stéphane Mallat. Separation and concentration in deep networks, 2021.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence?, 2019.

Yaofeng Desmond Zhong, Tongtao Zhang, Amit Chakraborty, and Biswadip Dey. A neural ode interpretation of transformer layers, 2022.

A Appendix and Supplementary material

A.1 Suite of Large Language Models and Prompt Data

We evaluate 38 total LLMs (24 base LLMs and 14 fine-tuned) on 6 datasets from the HuggingFace Open LLM leaderboard (Beeching et al., 2023).

Base models. Falcon (40B, 7B) (Almazrouei et al., 2023), Llama-3 (70B, 8B) (AI@Meta, 2024), Llama-2 (70B, 13B, 7B) (Touvron et al., 2023), MPT (30B, 7B) (Team, 2023a;b), Mistral v0.1 (Jiang et al., 2023), Gemma (7B, 2B) (Team et al., 2024), Gemma 1.1 (7B, 2B), NeoX (20B) (Black et al., 2022), Neo (2.7B) (Black et al., 2021; Gao et al., 2020), Pythia (6.9B).(Biderman et al., 2023), and GPT-2 (1.5B, 774M, 355M, 117M) (Radford et al., 2019).

Fine-tuned models. CodeLlama (34B, 13B, 7B) (Rozière et al., 2024), CodeLlama Instruct (34B, 13B, 7B) (Rozière et al., 2024), Mistral-v0.1 Instruct (7.3B) (Jiang et al., 2023), Mistral-v0.2 Instruct (Jiang et al., 2023), CodeGemma (Team et al., 2024).

Datasets. ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), Truthful QA (Lin et al., 2022), WinoGrande (Sakaguchi et al., 2019).

Table 1: **LLMs featured in the experiments throughout paper.** Included in the table is the parameter budget of each model, the embedding dimension, the number of training tokens, and the Open LLM leaderboard (Beeching et al., 2023) benchmark score.

MODEL	PARAM.	LAYERS (L)	DIM. (d_{MODEL})	TOKENS	SCORE	POS.ENCODING
LLAMA-3	70 B	80	8192	15 T		ROPE
	8 B	32	4096	15 T	62.35	ROPE
LLAMA-2	70 B	80	8192	2 T	66.05	ROPE
	13 B	40	5120	2 T	55.69	ROPE
	7 B	32	4096	2 T	50.97	ROPE
CODELLAMA	34 B	48	8192	2 T	55.33	ROPE
	13 B	40	5120	2 T	45.82	ROPE
	7 B	32	4096	2 T	39.81	ROPE
CODELLAMA (IT)	34 B	48	8192	2 T	43.0	ROPE
	13 B	40	5120	2 T	37.52	ROPE
	7 B	32	4096	2 T	40.05	ROPE
ORCA	13 B	40	5120	2 T	58.64	ROPE
	7 B	32	4096	2 T	54.55	ROPE
FALCON	40 B	60	8192	1 T	58.07	ROPE
	7 B	32	4544	1.5 T	44.17	ROPE
FALCON (IT)	40 B	60	8192	1 T	43.26	ROPE
	7 B	32	4544	1.5 T		ROPE
MPT	30 B	48	7168	1 T	66.98	ALIBI
	7 B	32	4096	1 T	56.83	ALIBI
PHI	2 B	32	2560	1.4 T	61.33	ROPE
	1.5 B	24	2048	150 B	47.69	ROPE
	1 B	24	2048	54 B		ROPE
MISTRAL-V0.1	7.3 B	32	4096		60.97	ALIBI
MISTRAL-V0.1 (IT)	7.3 B	32	4096		54.96	ALIBI
MISTRAL-V0.2 (IT)	7.3 B	32	4096		65.71	ALIBI
GEMMA	7 B	28	3072	6 T	64.29	ROPE
	2 B	18	2048	6 T	42.75	ROPE
GEMMA-1.1	7 B	28	3072	6 T	30.0	ROPE
	2 B	18	2048	6 T	60.09	ROPE
CODEGEMMA	7 B	28	3072	6 T	56.73	ROPE
	2 B	18	2048	6 T	32.19	ROPE
NEO	20 B	44	6144		41.69	ROPE
	2.7 B	32	2560	0.42	36.20	SINE
PYTHIA	12 B	36	5120	0.3 T	58.9	ROPE
	6.9 B	32	4096	0.3 T	39.30	ROPE
	2.8 B	32	2560	0.3 T		ROPE
	1.4 B	24	2048	0.3 T		ROPE
	1 B	16	2048	0.3 T		ROPE
GPT-2	1.5 B	48	1600		34.12	SINE
	774 M	36	1280		32.07	SINE
	355 M	24	1024		29.87	SINE
	117 M	12	768		28.53	SINE

A.2 ADDITIONAL METRICS

A.2.1 VISUALIZATION OF TRAJECTORIES WITH PCA

Each token, with initial embedding x_i^0, forms a trajectory $x_i^0, x_i^1, \ldots, x_i^L$ as it passes through the L transformer blocks. The dynamics in high-dimensional space are visualized through a 2-dimensional principal component (PC) projection, PC_L, fitted to the last layer embeddings $X^L = (x_1^L, x_2^L, \ldots x_n^L)$. The projected embeddings, $\text{PC}_L(x_i^0)$, $\text{PC}_L(x_i^1)$, $\ldots$, $\text{PC}_L(x_i^L)$, are plotted for each of the $i = 1, \ldots, n$ trajectories.

A.3 COMPARISON TO RESNETS

Coupling. Our results complement and build upon those of (Li & Papyan, 2023), who have observed the coupling of singular vectors of Residual Jacobians in classification ResNets. We observe coupling across depth in a wide range of LLMs, where Jacobians are evaluated at the sequence of embeddings, with respect to the current token, whereas in RA Jacobians are evaluated at a single representation. Additionally, with transformers we may analyze coupling not only across depth but also across tokens. We observe coupling in LLMs across tokens in a variety of ways. Further, we consider and analyze the relationship between coupling and generalization.

Linearity and Equidistance. Linearity in hidden trajectories, as observed in ResNets Li & Papyan (2023); Gai & Zhang (2021), also emerges with training in LLMs. The mean LSS value among the evaluated LLMs is 4.24 (Figure 5(a)), greater than LSS measurements observed for ResNets (Gai & Zhang (2021), page 18) which range between 2.0-3.0 (due to varying trajectory length and hidden dimension). In both architectures, training induces improved linearity and regularity (Figure 12) in trajectories. In contrast to ResNets, trajectories are not equidistant, instead showing exponential growth between layers (Figure 5(b)). We quantify this spacing through a low coefficient of variation, displaying the presence of exponential growth in token trajectories. In both classifier ResNets and LLM transformers, there is an evident level of regularity in hidden representations (Figure 12).

Rank of Jacobians. Li & Papyan (2023) show that residual Jacobians have rank at most C, the number of classes. This analogous result automatically holds for LLMs since the vocabulary size is significantly greater than the embedding dimension of the transformer blocks.

Singular Value Scaling. Li & Papyan (2023) observe that top singular values of residual Jacobians scale inversely with depth. In trained LLMs, however, top singular values do not show a consistent depth scaling across models (Figure 34), notably differing from classification ResNets. In addition, the distribution of singular values at each layer varies significantly between models. Singular value scaling is more present in untrained transformers (Figure33), likely caused by additional layer normalizations in residual blocks (Section 4.1).

A.4 Significance of Coupling

The transformer block coupling phenomenon offers insight into several prominent practices in LLM research, as summarized in Table 2.

A.5 Dynamical Motivation

The equality of top left and right singular vectors suggests that the linearizations form a simple linearized system that acts on representations. Consider a difference equation

$$x^{l+1} - x^l = A_l \, x^l \tag{19}$$

Its solution at the final L is given by

$$x^L = \prod_{l=1}^{L} (I + A_l) \, x^0 \tag{20}$$

Expanding the brackets shows that x^l can be thought of as a collection of many paths of various lengths, due to the binomial identity. This agrees with Veit et al. (2016) which views ResNets as

$$x^l = (I + A_{l-1}) \, (I + A_{l-2}) \dots (I + A_1) \, x^0 \tag{21}$$

However, Veit et al. (2016) do not make any assumptions about the alignment of the various A_l matrices. The coupling phenomenon suggests the model as implementing the simpler system

$$x^l = (I + A)^l \, x^0 \tag{22}$$

where all the A matrices are aligned. One benefit of this interpretation is that, we can write x^l in a simple closed form, as above. We quantify the similarity of hidden trajectories to the evolution of the above difference equation, in order to detect the emergence of a simple linearization to representations. The emergence of increased linearity and exponential spacing in many LLMs can be analyzed as a result of coupling under some conditions on the spectral decomposition of the Jacobians. Considering input embeddings $x_1^0, \dots, x_n^0$ and the linearization of the last token embedding x_n^l given by $J_{n,n}^l(x_1^0, \dots, x_n^0)$:

$$x_n^l = (I + J_{n,n}^l(x_1^0, \dots, x_n^0)) x_n^{l-1} \tag{23}$$

Table 2: **Significance of the Coupling Phenomenon.** A table which highlights the implications of transformer block coupling to a variety of effors in machine learning research.

Current Research Practice	Key Idea	Our Contribution
Compressing models by merging blocks (Fu et al., 2022; Kim et al., 2024)	Combine adjacent transformer blocks to reduce model size without significant performance loss.	Demonstrates that merging is effective because blocks become strongly coupled during training.
Compressing models by pruning blocks (Elkerdawy et al., 2020; Kim et al., 2024; Dror et al., 2021; Fang et al., 2023)	Remove certain transformer blocks while preserving functionality.	Explains that pruning works because the coupling ensures redundancy across blocks.
Compressing models by projecting weight matrices (Ashkboos et al., 2024)	Reduce dimensionality by projecting weights into smaller subspaces.	Shows that coupling induces a low-dimensional subspace in which blocks' weights are aligned.
Studying the effect of transformer block permutations (Hu et al., 2021; Mahabadi et al., 2021; van der Ouderaa et al., 2024; Li et al., 2016)	Investigate whether permuting the order of blocks affects model performance.	Explains why permutations have minimal impact: strong coupling creates structural robustness.
Early exiting in LLMs (Scardapane et al., 2020; Jazbec et al., 2024)	Allow models to exit computation early based on task confidence.	Reveals that early exiting works because representations progress linearly along a shared trajectory due to coupling.

We simplify notation and write $x_n^l = x^l$, $J_{n,n}^l(x_1^0, \ldots, x_n^0) = J^l$. Under the assumption of spectral coupling, $J^l = U_l S_l V_l^T \approx U S_l U^T$, and the linearized effect of the last token is

$$x^L = \prod_{l=1}^{L}(I + U_l S_l V_l^T)x^0 \approx U\left(\prod_{l=1}^{L}(I + S_l)\right)U^T x^0 \tag{24}$$

Suppose that $x^0 = u_k$ is the k-th left singular vector of J^l. It follows that $x^L = \prod_{l=1}^{L}(1 + s_k^l)x^0$, where s_k^l denotes the k-th singular value at layer l. The exponential spacing measurement is motivated by the consistent choice $s_k = s_k^1 = s_k^2 = \cdots = s_k^L$. Explicitly, $x^L = (1 + s_k)^L x^0$, and by Equation 17, for each l

$$\alpha_k^l = \ln\left(\frac{(1 + s_k)^l ||u_0||}{(1 + s_k)^{l-1}||u_0||}\right) = \ln(1 + s_k) \implies \text{ED} = 0$$

that is, the coefficient of variation 0 across l. In addition, if $x^l = (1 + s_k)x^{l-1}$, it is clear that the trajectory would form a perfect line, yielding $LSS = 1$ by the discussion in Section 3.2. In general, trajectories are not expected to be perfectly linear unless x^0 aligns with an eigenvector of J^l.

A.6 LLM Evaluation and Implementation Details

The source code used to produce the results reported in this experiment has been included as supplemental material. Models with varying parameter sizes are loaded on GPUs with appropriate memory requirements: NVIDIA A40 ($n_{\text{param}} \geq 40B$), NVIDIA Quadro RTX 6000 for Gemma variants and when ($40B > n_{\text{param}} > 13B$), and NVIDIA Tesla T4 when ($13B \geq n_{\text{param}}$) except Gemma variants. 1,200 prompts from the OpenLLM leaderboard were evaluated in variable batch sizes were queued on a SLURM cluster, with appropriate adjustments depending on the memory required to load the LLM.

- $13B \geq n_{\text{param}}$: 100 prompts per batch, except Gemma variants, which used 25 prompts per batch. The larger memory requirement for Gemma variants is likely due to the much larger vocabulary size in the model.
- $40B > n_{\text{param}} > 13B$: 10 prompts per batch, except NeoX 20B which used 100 prompts per batch.
- $n_{\text{param}} \geq 40B$: 50 prompts per batch.

Due to the high memory requirement for computing Jacobians, for experiments involving the Jacobians, NVIDIA Quadro RTX 6000 was used additionally for $13B > n_{\text{param}} \geq 7B$ and corresponding models were quantized. Additionally, due to compute restrictions, alignment of singular vectors of Jacobians was computed on a smaller subset of the 1,200 prompts.

The computational complexity for the metrics utilized in this paper are as follows:

Coupling. Coupling requires computing the Jacobians for each transformer block, and so a forward pass and backward pass required (note that we compute the Jacobians on a block level). Once the Jacobians are obtained, it requires computing a truncated singular value decomposition of each Jacobian. The time complexity of computing the truncated SVD of rank k for a $d \times d$ matrix is $\mathcal{O}(d^2k)$, where $k << d$. Computing A from the SVDs then has time complexity $\mathcal{O}(k^3)$, so the asymptotic time complexity of computing the coupling score between two connections is $\mathcal{O}(d^2k)$.

LSS. For each trajectory, the time complexity of computing the LSS is $\mathcal{O}(Ld)$ where L is the number of layers and d is the hidden dimension. Therefore, for a prompt containing T tokens, the total time complexity for each prompt is $\mathcal{O}(TLd)$ (in addition to a single forward pass of the model).

Expodistance. Similarly, computing the expodistance of a single trajectory has time complexity $\mathcal{O}(Ld)$. Therefore, for a prompt containing T tokens, the total time complexity for each prompt is $\mathcal{O}(TLd)$ (in addition to a single forward pass of the model).

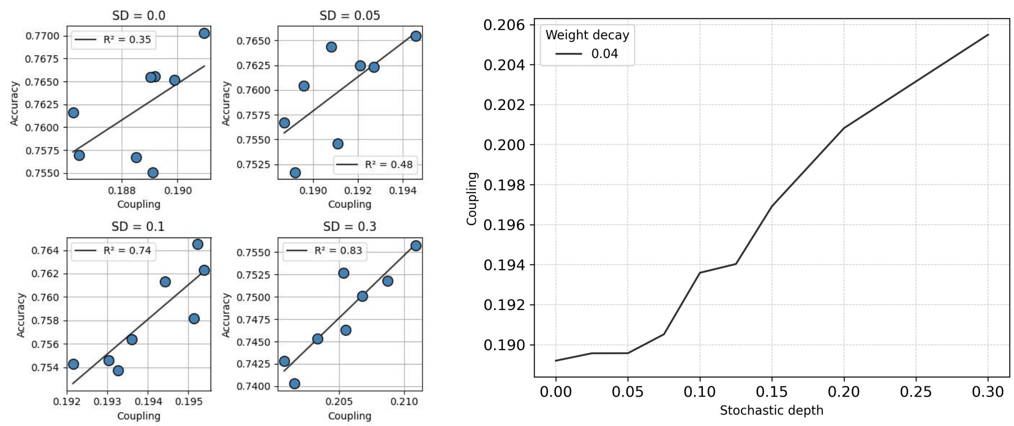
A.7 ViT Training Details

For further investigation of coupling in transformers, we train 64 Vision Transformers (ViTs) following the default configurations of DEiT training (Touvron et al., 2021) on CIFAR10 (Krizhevsky, 2009). For a fixed ViT architecture with embedding dimension 192, depth of 12 layers, and 3 attention heads, we vary the weight decay $\{0.005, 0.01, 0.02, 0.04, 0.06, 0.08, 0.1, 0.2\}$ and stochastic depth rate $\{0, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2, 0.3\}$. Optimization uses ADAM optimizer with $5e-4$ learning rate and cosine scheduler for 500 epochs with 30 epochs of linear warmup. Training proceeds with data mixup using $\alpha = 0.8$ Optimization proceeds on 4 NVIDIA Tesla T4 GPUs with 128 batch size and data parallelization, with total training time being approximately 2 hours.

A.8 Additional Experimental Results

Table 3: R^2 for Various Models

Model	$\boldsymbol{R^2}$
Llama2-7B	0.5484
MPT-30B	0.9629
Gemma-7B	0.7832
Mistral-v0.1 7B	0.5876



(a) (b)

Figure 6: **Transformer Block Coupling in ViTs.** **(a)** CIFAR10 test accuracy against normalized coupling for stochastic depths $(0.0, 0.05, 0.1, 0.3)$ trained with varying weight decay values (Section 3.4). Coupling and accuracy have correlation $R^2 = (0.35, 0.48, 0.74, 0.83)$ that respectively increase with the SD rate. **(b)** Coupling against stochastic depth rate among ViTs trained with weight decay 0.04, and shows an increase with stochastic depth. Please see Figures 10, 11 for further details.

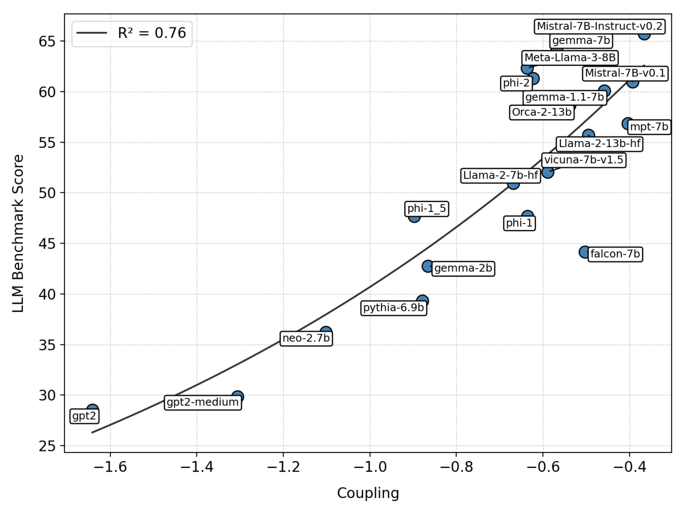


Figure 7: Exponential fit between coupling and performance.

Subset	R^2	p**-value**
All models	0.75	1.56×10^{-6}
7B models only	0.55	0.023
Score > 45	0.39	0.023

Table 4: Summary of R^2 and p-values for different subsets of models for the correlation between coupling and performance.

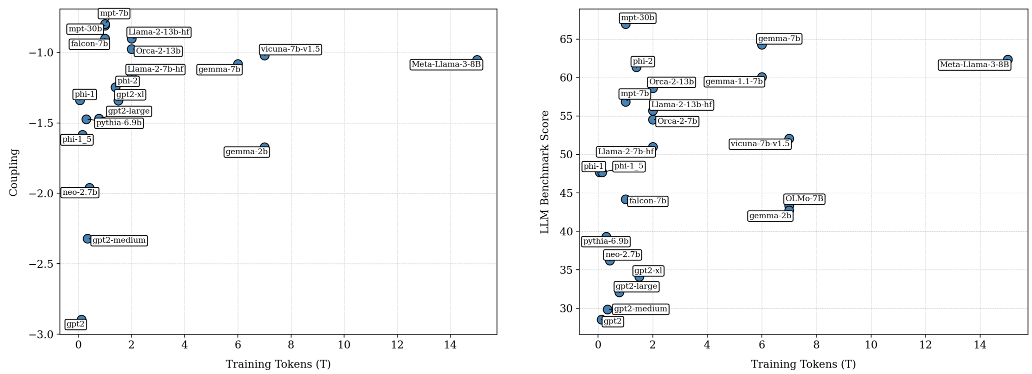


Figure 8: Plotting (1) the number of tokens used to train each model against its mean depth-wise coupling score against and (2) the number of training tokens against its LLM Huggingface Benchmark score.

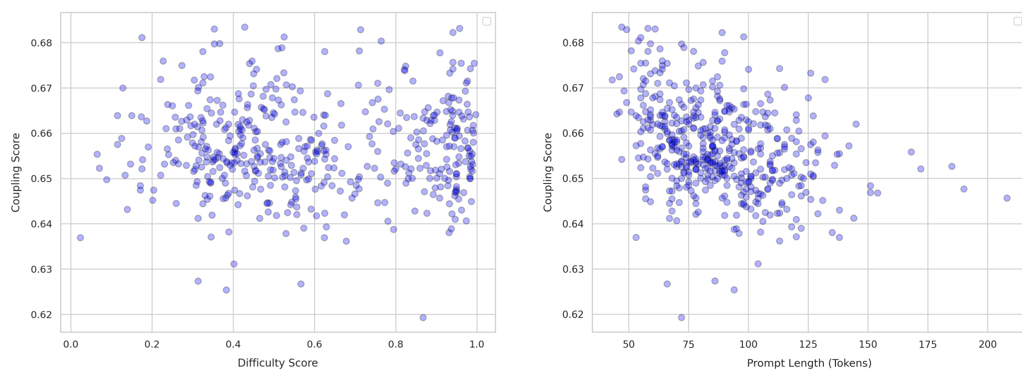


Figure 9: Utilizing the Easy2Hard Benchmark dataset Ding et al. (2024), the difficulty score of a prompt is plotted against the mean depth-wise coupling over that prompt. Additionally, plotted on the right is the prompt length (in tokens) plotted against the mean depth-wise coupling on that prompt.

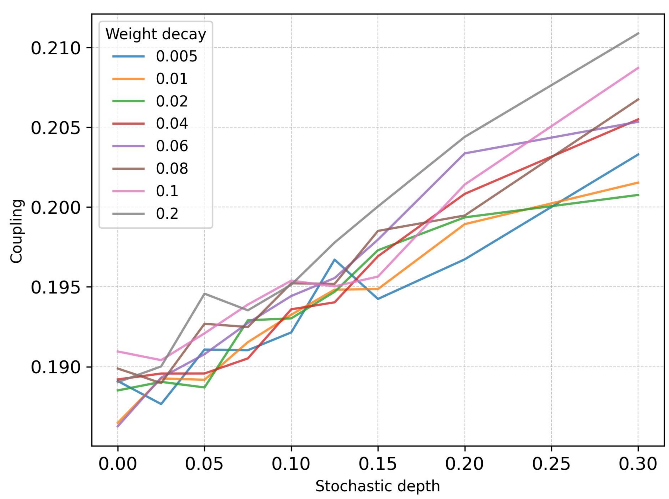


Figure 10: **Coupling against Stochastic Depth Rate.** Plots are generated for each weight decay in $\{0.005, 0.01, 0.02, 0.04, 0.06, 0.08, 0.1, 0.2\}$.

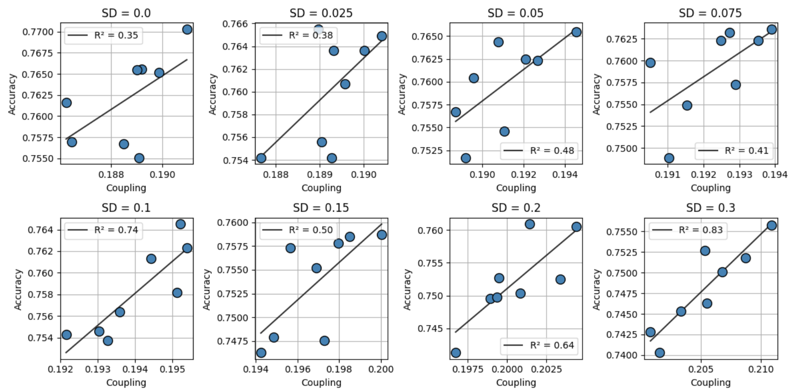
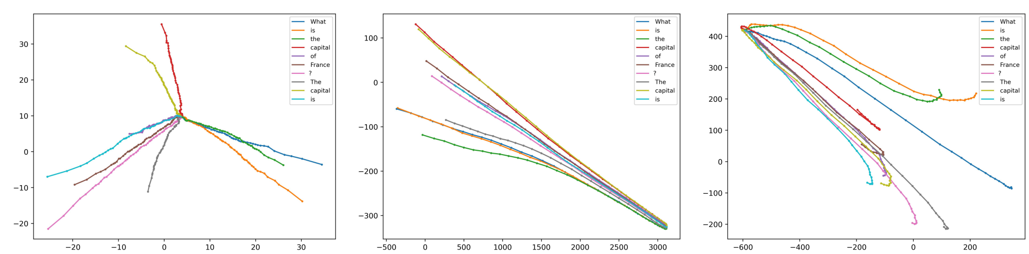


Figure 11: **Transformer Block Coupling for Fixed Weight Decay.** Coupling plotted against accuracy for 64 ViTs with stochastic depths $\{0, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2, 0.3\}$.



(a) Llama-3 70B (b) MPT 30B (c) NeoX 20B

Figure 12: **Trajectories of Hidden Representations.** Visualization of the layer-wise trajectories of hidden representations in Llama 3 70B, MPT 30B, and NeoX 20B in the prompt: `What is the capital of France? The capital is`. Trajectories of tokens are plotted in latent space, visualized with a 2-dimension principal component projection. A clear directed and outward growth is visible in each token trajectory.

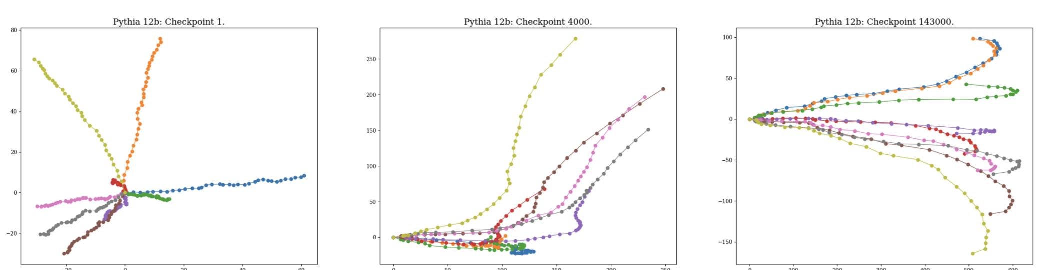


Figure 13: **Evolution of Hidden Trajectories Throughout Training.** Principle component visualizations of the hidden trajectories in Pythia 12B at training checkpoints 1, 4000 and 143000 on the prompt: `What is the capital of France? The capital is.`

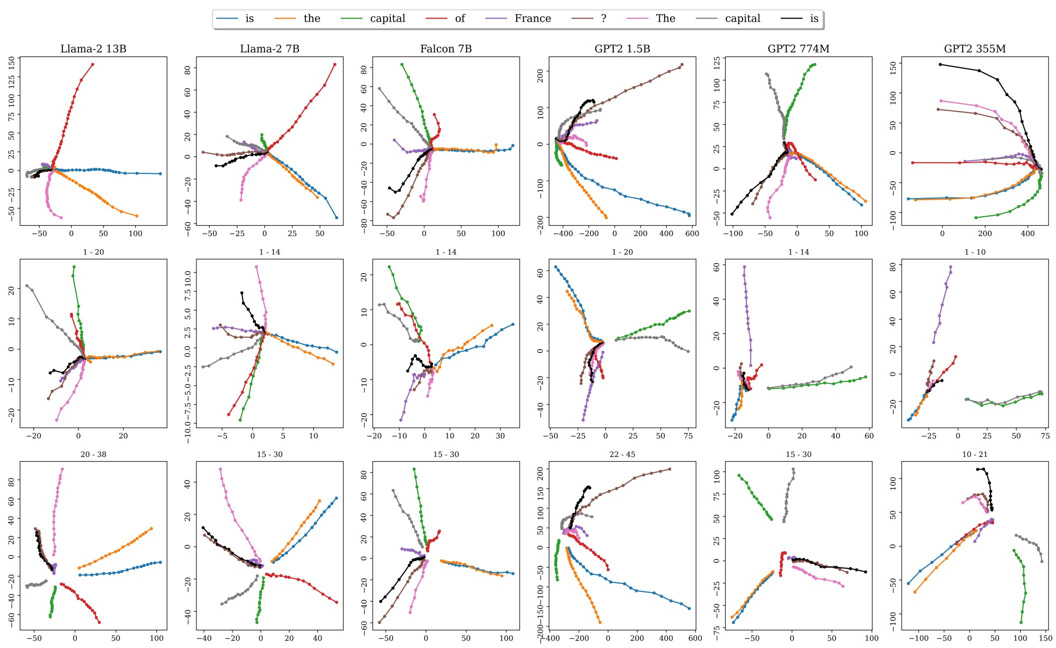
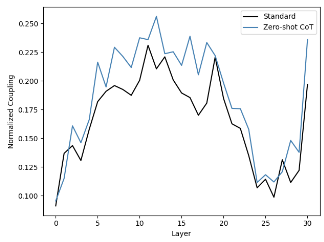
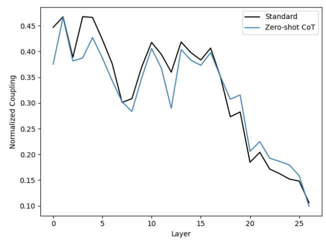


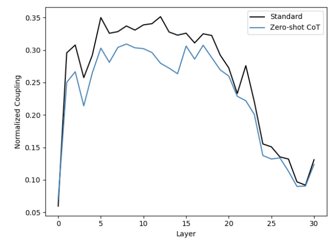
Figure 14: **Hidden Trajectories in LLMs.** Principal components of the trajectories of the hidden representations through various LLMs (columns, decreasing in model size, see Table 1) in the prompt: `What is the capital of France?  The capital is`. Top row: all layers. Middle Row: layers in shallower transformer blocks (layers specified above plot). Bottom Row: layers in deeper transformer blocks (layers specified above plot). Trajectories of each input token (last token 'is' is plotted in black) are plotted in latent space, visualized with a 2-dimension principal component projection. Representations proceed in distinct outward directions, especially in the second half of transformer blocks (lower row) during which the norm of representations increases, with possible abrupt change in the last layer (outer points in upper row). A clear direction of movement is visible in each token trajectory.



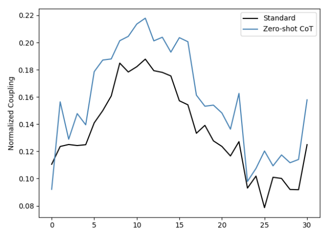
(a) Llama-3 8B



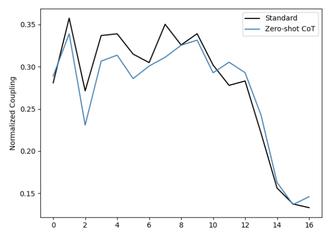
(b) Gemma 7B



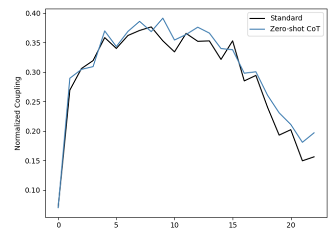
(c) Phi-2



(d) Llama-2 7B

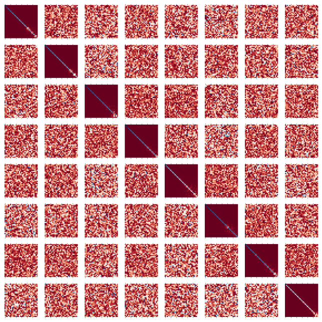


(e) Gemma 2B

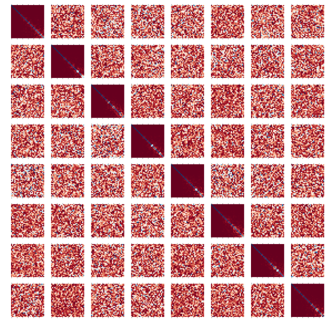


(f) Phi-1.5

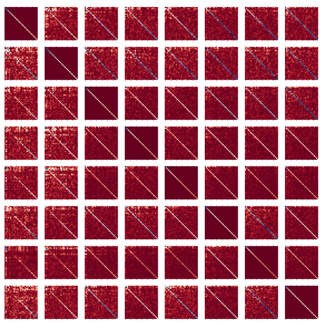
Figure 15: **Zero-shot Chain of Thought on Depth-wise Coupling across Layers**. We compare the normalized coupling on prompts from GSM8k with that of the same prompts appended with ”Let’s think step by step.”, which we refer to as the Zero-Shot CoT (Kojima et al., 2023) prompts. For a more thorough analysis, we measure how much each layer is coupled with all other layers, as shown in the figures above. Firstly, the coupling across layers exhibits distinct behaviors across different models, but with noticeable similarities within the same model families In the LLaMA models, coupling starts off lower, increases in the middle layers, then decreases before showing a slight increase again at the final layers. In the Gemma models, coupling begins relatively high and steadily decreases toward the end of the network. In contrast, the Phi models exhibit significantly lower coupling in the first layer, followed by an immediate increase, and then a slight decrease in coupling toward the final layers. The CoT prompt produces similar coupling patterns to the standard prompt, with slight variations in coupling strength. Specifically, in the LLaMA models, the CoT prompt consistently results in higher coupling across layers. For the Gemma models, the CoT prompt leads to similar overall coupling levels, though some layers exhibit slightly lower coupling and others slightly higher. On the other hand, Phi-2 shows consistently lower coupling with the CoT prompt, while Phi-1.5 is marginally higher. This variability in behavior, along with the similarities within model families, is likely due to differences in training methods and data across organizations, while models within the same family are trained with potentially similar methodologies.



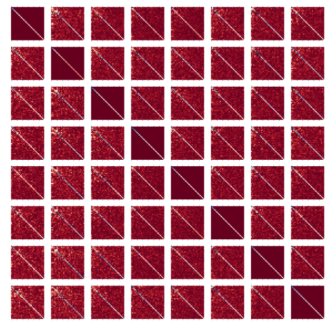
(a) Gemma 7B **Untrained**



(b) Phi-2 **Untrained**

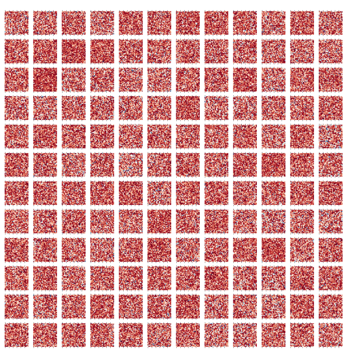


(c) Gemma 7B **Trained**

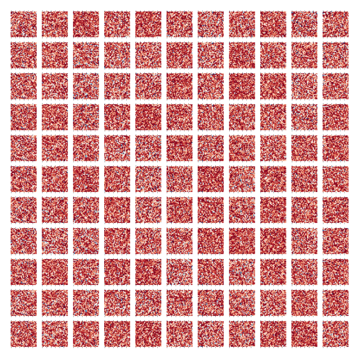


(d) Phi-2 **Trained**

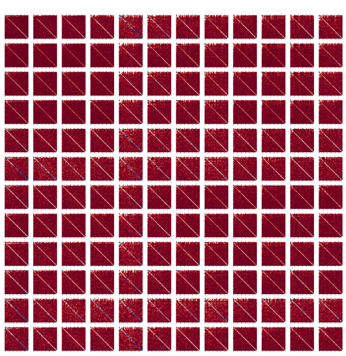
Figure 16: **Transformer Block Coupling across Depth**.



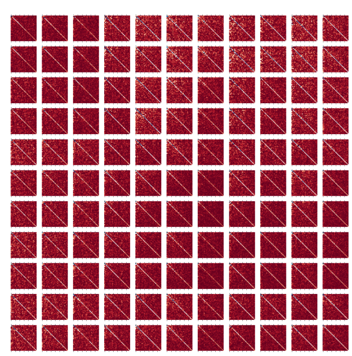
(a) Gemma 7B **Untrained**



(b) Phi-2 **Untrained**

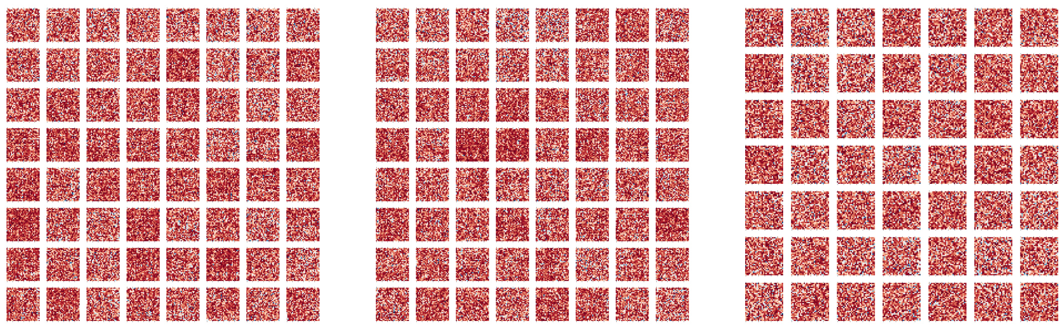


(c) Gemma 7B **Trained**

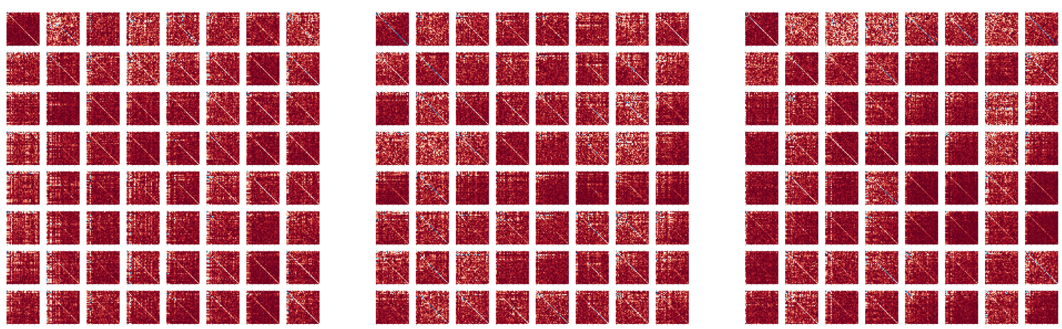


(d) Phi-2 **Trained**

Figure 17: **Transformer Block Coupling across Tokens (Same input and output tokens)**.

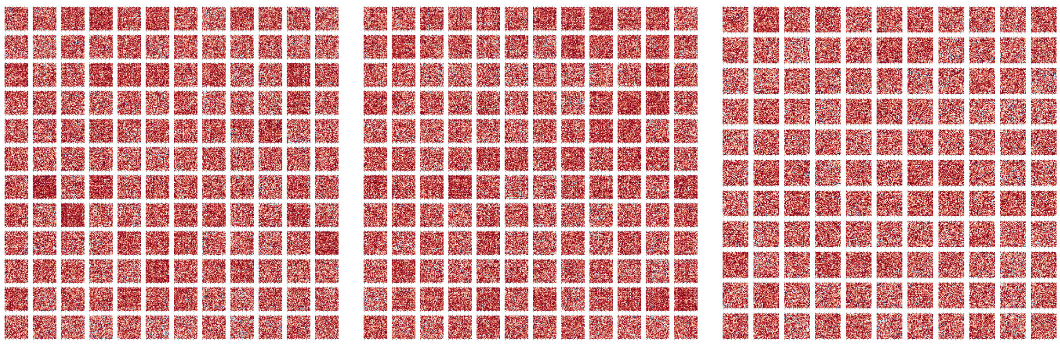


(a) Llama-3 8B **Untrained** (b) Gemma 7B **Untrained** (c) Phi-2 **Untrained**

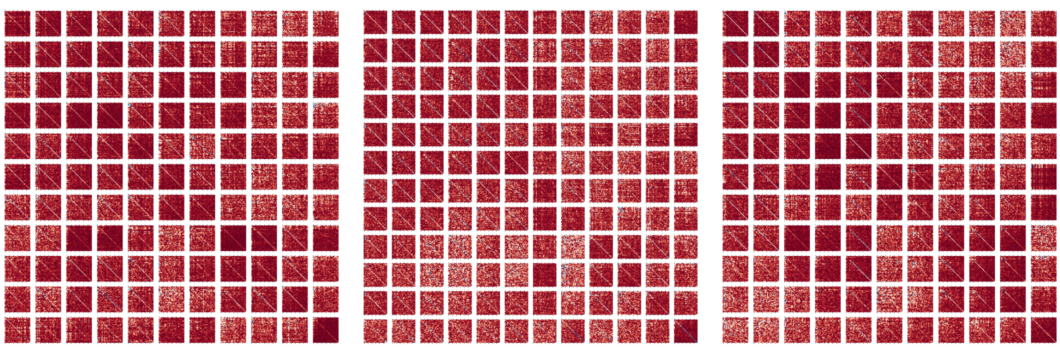


(d) Llama-3 8B **Trained** (e) Gemma 7B **Trained** (f) Phi-2 **Trained**

Figure 18: **Transformer Block Coupling across Token (Fixed input)**. The figure illustrates coupling of Jacobians, with fixed input token, across tokens. More specifically, in the matrix plot located at entry (t_2, t_2'), the absolute values of the entries of matrices $A_{ll'}^{t_1 t_2 t_1 t_2'}$ are visualized (with randomly fixed layers l, l'). In the trained plots (bottom row), the off-diagonal entries being close to 0 with visible diagonal indicates coupling of these Jacobians. This coupling, however, seems to be more evident for certain token pairs and less for others. At initialization (top row), there is no such coupling across tokens. Additional visualizations are included in Appendix A.8 (Figure 22)

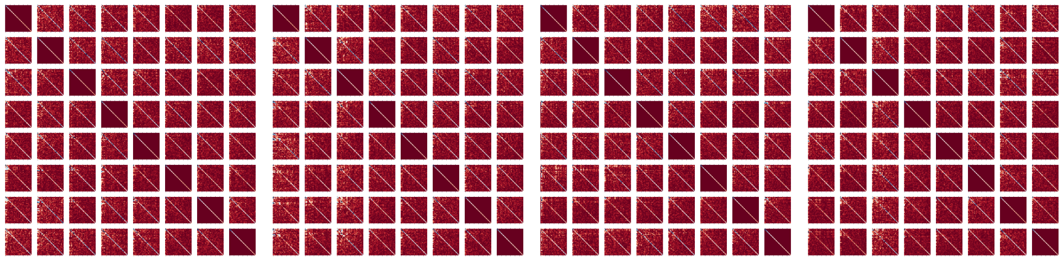


(a) Llama-3 8B **Untrained** (b) Gemma 7B **Untrained** (c) Phi-2 **Untrained**

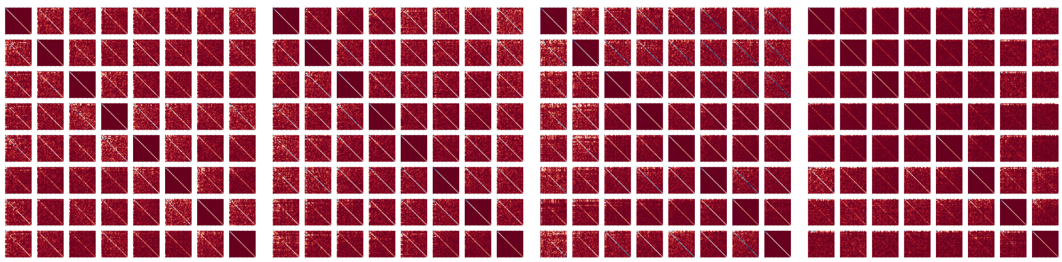


(d) Llama-3 8B **Trained** (e) Gemma 7B **Trained** (f) Phi-2 **Trained**

Figure 19: **Transformer Block Coupling across Token (Fixed output)**. The figure illustrates coupling of Jacobians, with fixed output token, across tokens. More specifically, in the matrix plot located at entry (t_1, t_1'), the absolute values of the entries of matrices $A_{ll'}^{t_1 t_2 t_1' t_2}$ are visualized (with randomly fixed layers l, l'). In the trained plots (bottom row), the off-diagonal entries being close to 0 with visible diagonal indicates coupling of these Jacobians. This coupling again seems to be more evident only for certain token pairs. At initialization (top row), there is no such coupling across tokens. Additional visualizations are included in Appendix A.8 (Figure 23)

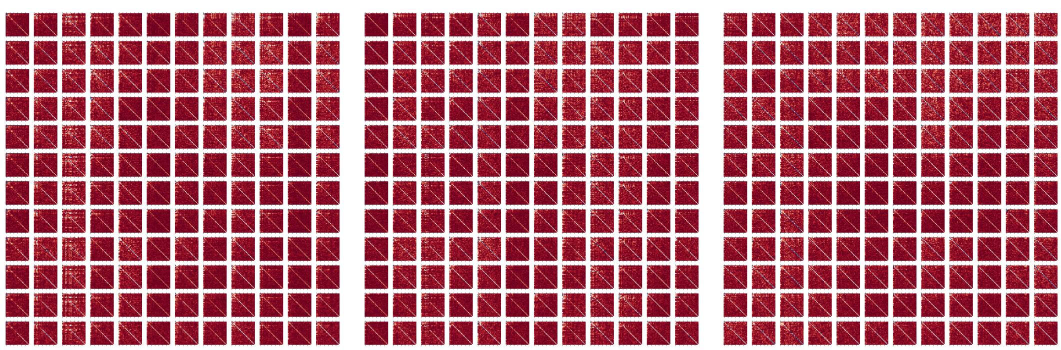


(a) Llama-2 13B (b) Llama-2 7B (c) Orca-2 7B (d) Vicuna 7B

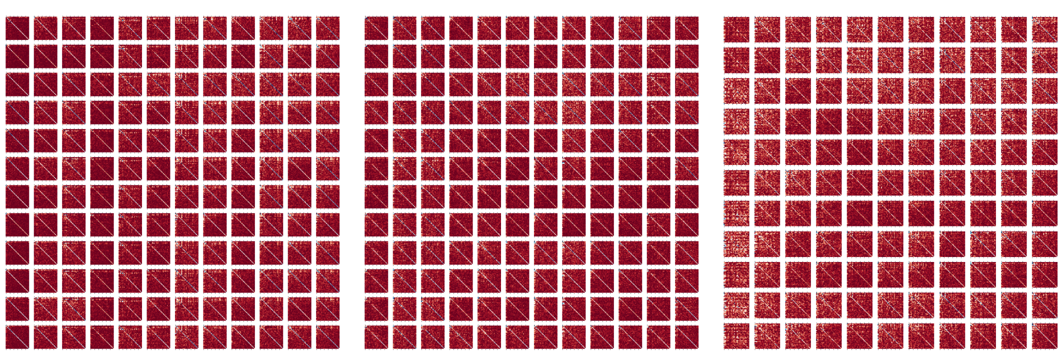


(e) Falcon 7B (f) Mistral 7B (g) Gemma 7B (h) Gemma 2B

Figure 20: **Additional plots of Coupling across depth**. The figure illustrates the alignment of Residual Jacobians across transformer blocks 9 to 16.

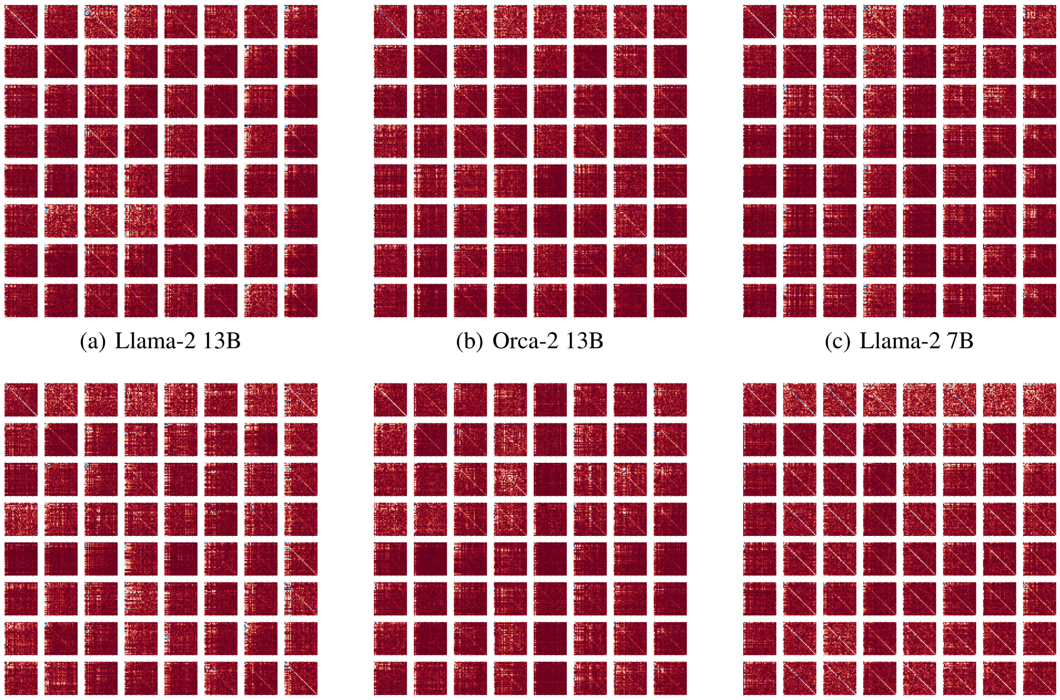


(a) Llama-2 13B (b) Orca-2 13B (c) Llama-2 7B



(d) Orca-2 7B (e) Vicuna 7B (f) Falcon 7B

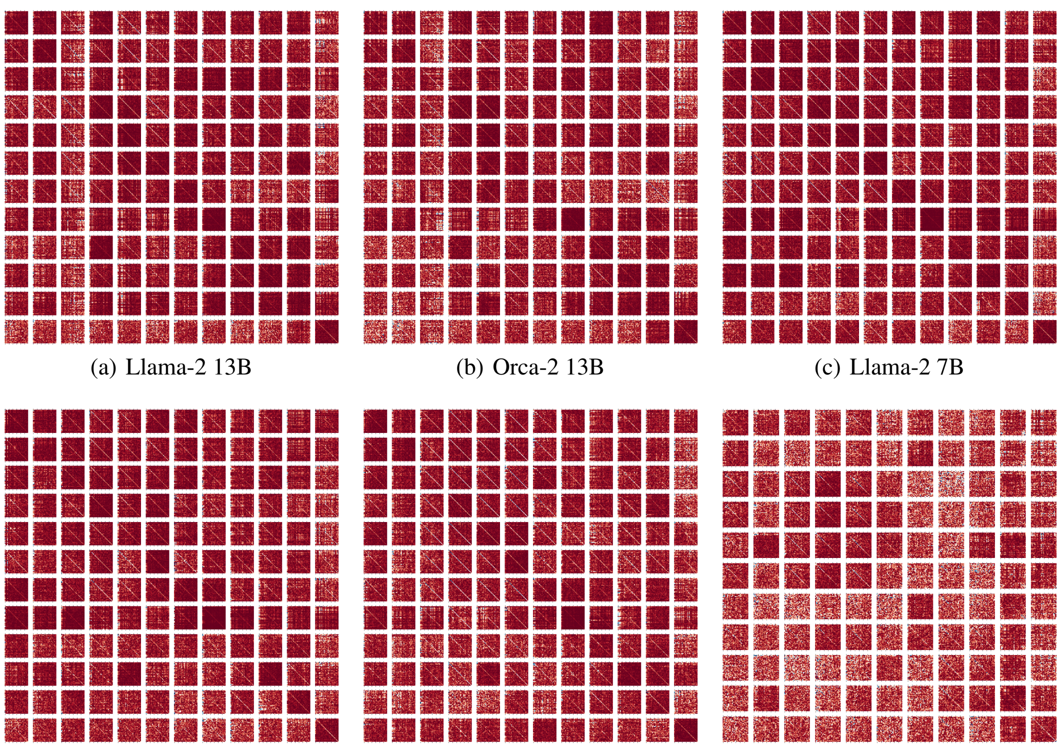
Figure 21: **Additional plots of Coupling across Tokens (same input and output tokens)**.



(a) Llama-2 13B (b) Orca-2 13B (c) Llama-2 7B

(d) Orca-2 7B (e) Vicuna 7B (f) Falcon 7B

Figure 22: **Additional plots of Coupling across Tokens (fixed input)**.



(a) Llama-2 13B (b) Orca-2 13B (c) Llama-2 7B

(d) Orca-2 7B (e) Vicuna 7B (f) Falcon 7B

Figure 23: **Additional plots of Coupling across Tokens (fixed output)**.

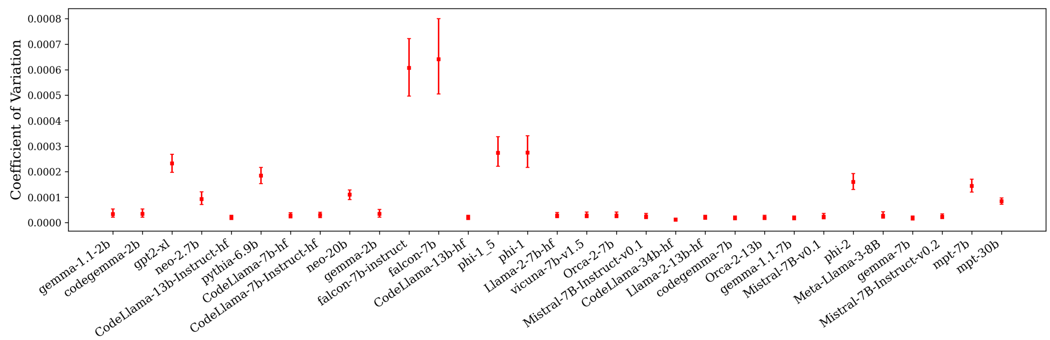


Figure 24: **Coefficient of variation of layer-wise equidistance.** Variation of layer-wise equidistance (Section 3.3) computed over 1,200 prompts from the HuggingFace Open LLM Leaderboard datasets (Section 4.2) on a suite of untrained LLMs (Appendix A.1). Plotted are the median values over all prompts, and are accompanied with uncertainty intervals depicting the inter-quartile range of the results for each model. The models are sorted by increasing benchmark performance.

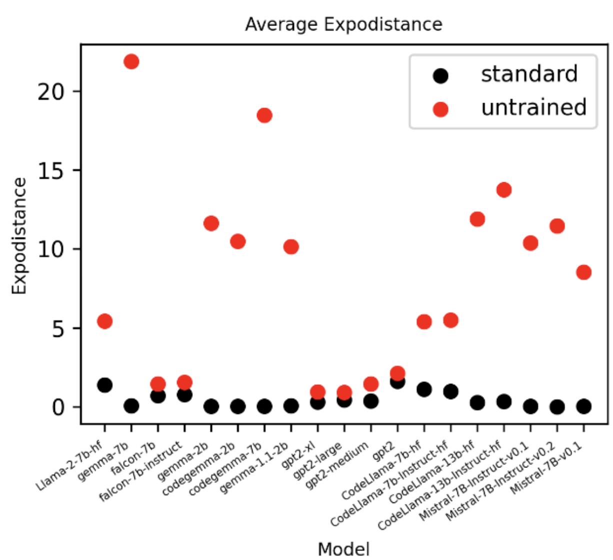


Figure 25: **Coefficient of variation of layer-wise expodistance.** Variation of layer-wise expodistance (Section 3.3) computed over 1,200 prompts from the HuggingFace Open LLM Leaderboard datasets (Section 4.2) on a suite of untrained LLMs (Appendix A.1).

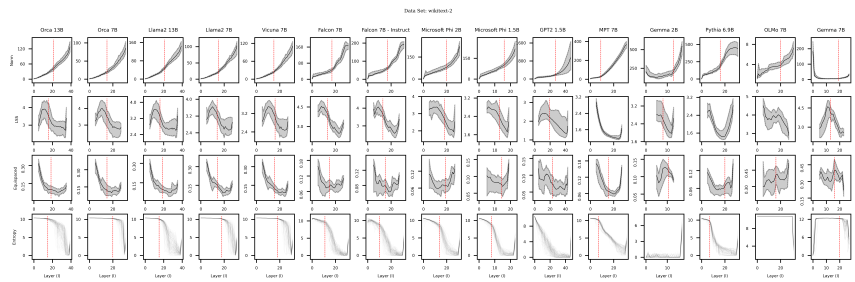


Figure 26: **Various Measurements of Representations.** All measurements were made on 100 prompts taken from the WikiText 2 datsets. (**Row 1**) Norm of hidden representations as a function of layer depth. (**Row 2**) Line Shape Score (LSS) of the hidden trajectories as a function of layer depth. (**Row 3**) Mean equidistance of contiguous hidden trajectories as a function of depth. (**Row 4**) Entropy of logit vectors as a function of depth. Noted in most plots is a line where the behaviour of the measurement drastically changes.

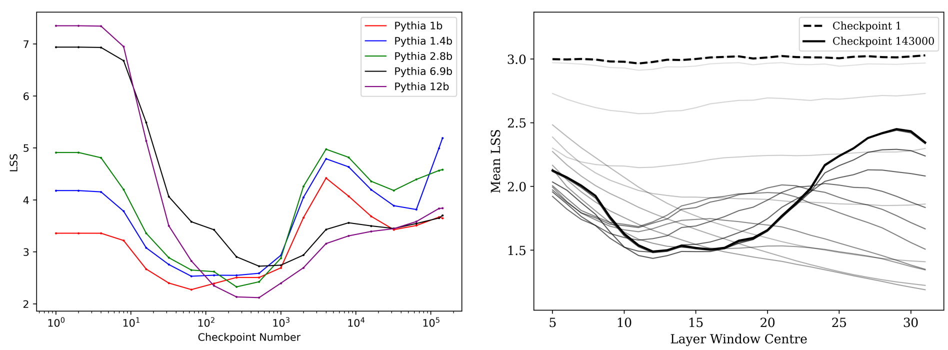
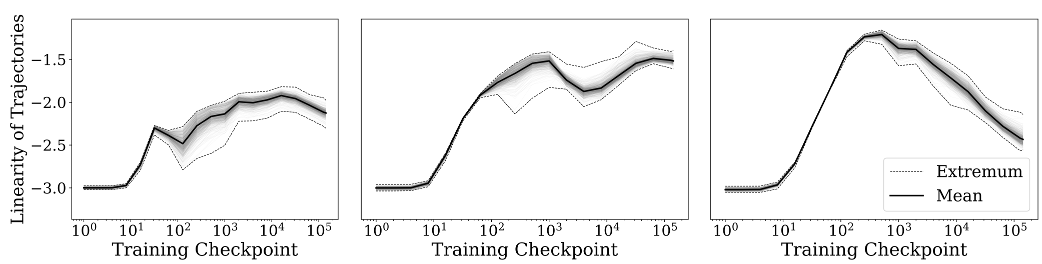
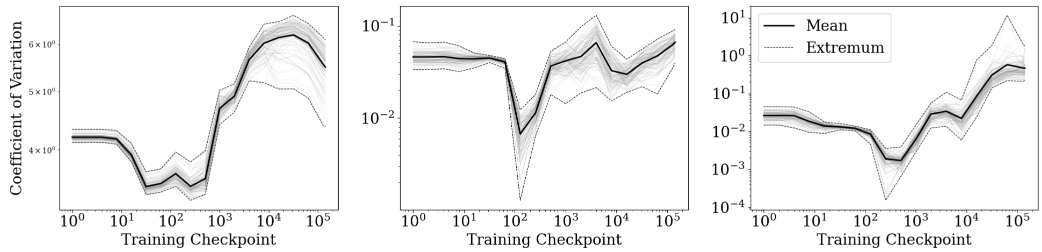


Figure 27: **Linearity Emerges with Training.** Two plots displaying the evolution of the linearity of the token trajectories through training. (**Left**) The LSS as a function of training checkpoint for the variants of the Pythia Scaling Suite Biderman et al. (2023). Here, the LSS is measured over each entire prompt. (**Right**) The mean LSS as a function of layer depth measured at various checkpoints throughout the Pythia 12B model. Here, the LSS is computed on a window of layers of width 11, centred at the value given by the x-axis.



(a) Layer $l = 5$ (b) Layer $l = 15$ (c) Layer $l = 30$

Figure 28: **Emergence of Linearity with Training.** Average linearity of a trajectory at block depths $l \in \{5, 15, 30\}$ evaluated for Pythia 12B (Biderman et al., 2023) checkpoints $\{1, 2, 4, \ldots, 256, 512, 1k, 2k, 4k, \ldots, 128k, 143k\}$. The linearity is given by the negative LSS, and is computed on a window of 11 layers centered at each depth l.



(a) Layer 5 (b) Layer 15 (c) Layer 25

Figure 29: **Expodistance at Fixed Layers.** Plotted are mean expodistances as a function of training checkpoint at various depths of the network. The values at a given depth are the mean expodistance over a layer window of width 11 centred at said depth 100 MMLU prompts are plotted at each layer.

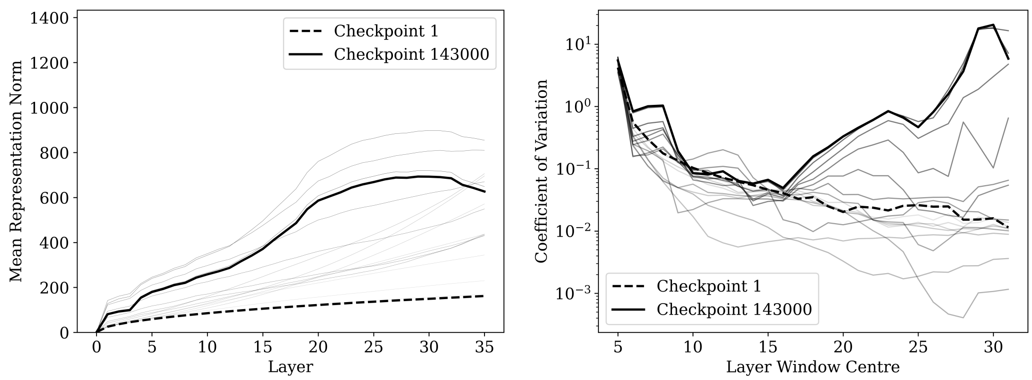


Figure 30: **Norm and Expodistance During Training.** (**Left**) Plotted is the norm of the representations as a function of depth at various training checkpoints. Observed is the transition form log-like growth in early stages to exponential-like growth, particularly through layers 5 through 20, as training evolves. (**Right**) Plotted is the expodistance over a layer window of width 11 centred at the give depth, each computed at a variety of training checkpoints.

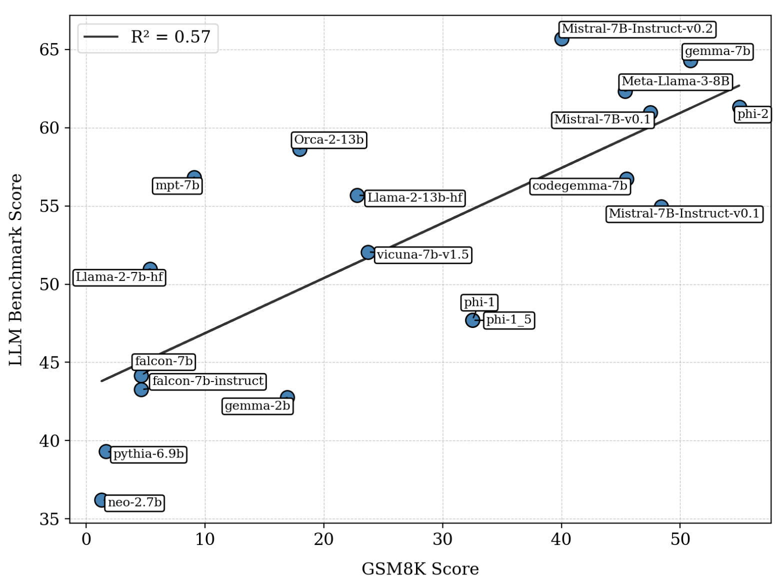


Figure 31: The score on GMS8K Cobbe et al. (2021) against the cumulative Huggingface LLM Benchmark score.

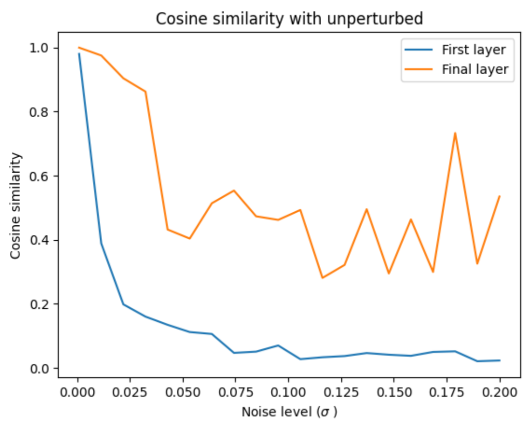
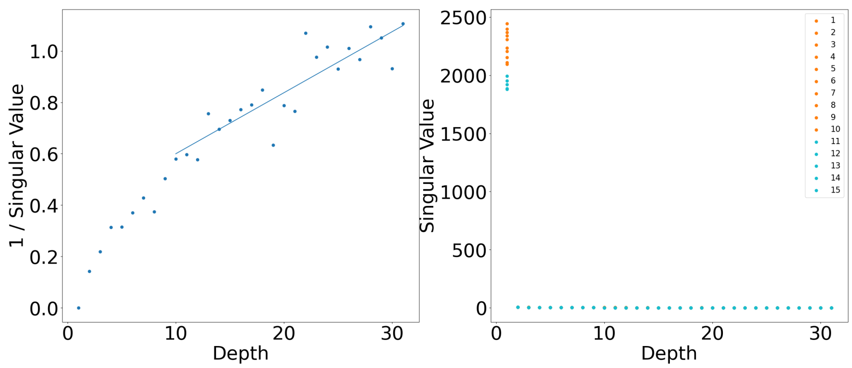
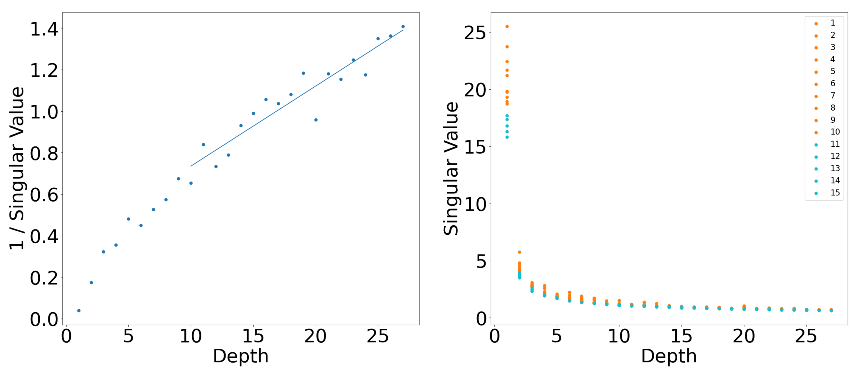


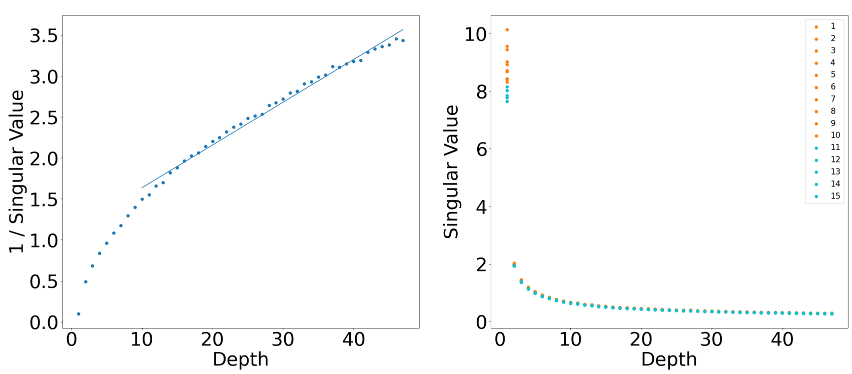
Figure 32: **Perturbation Experiments with Llama-3 8B.** The last token embedding is perturbed with various noise levels, and compared with the true embedding at the first and last layers. The trend shows that at small noise levels, cosine similarity with the true embedding remains somewhat high, and is significantly lower at the first layer.



(a) Llama-3 8B

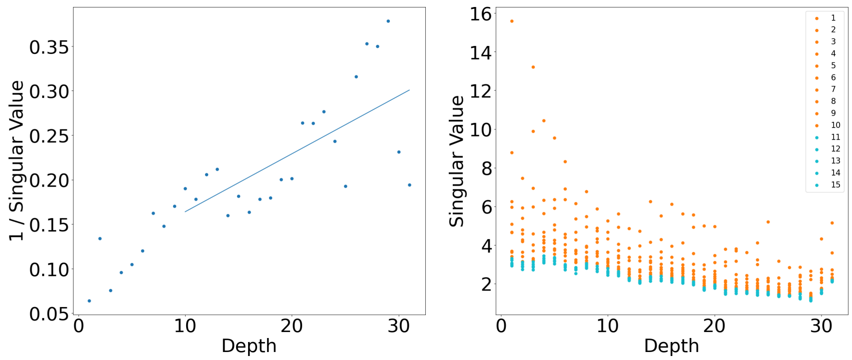


(b) Gemma 7B

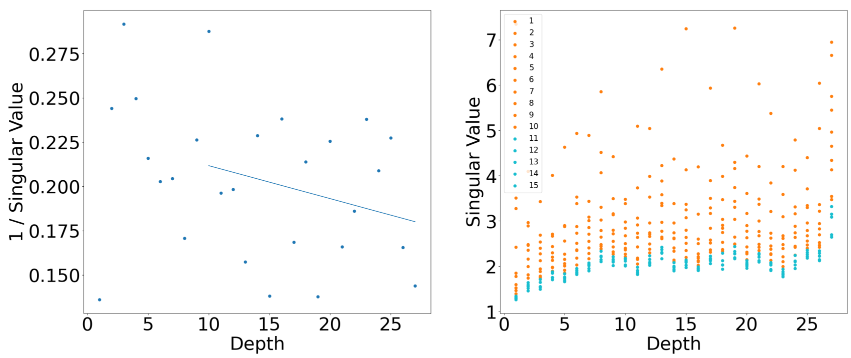


(c) GPT-2 XL

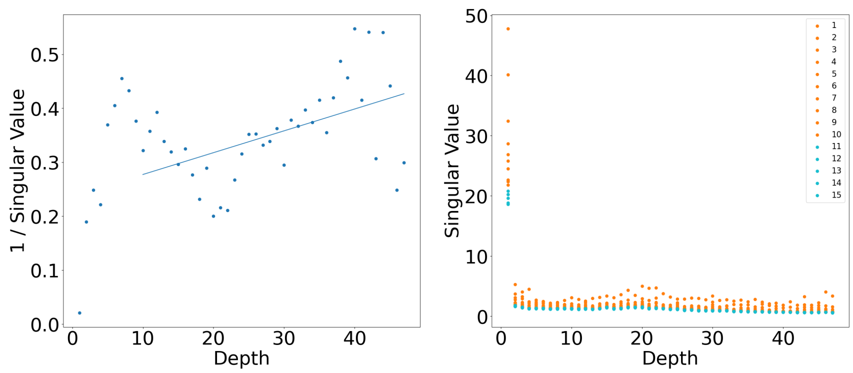
Figure 33: **Scaling of Singular Values of Residual Jacobians (Untrained)**.



(a) Llama-3 8B

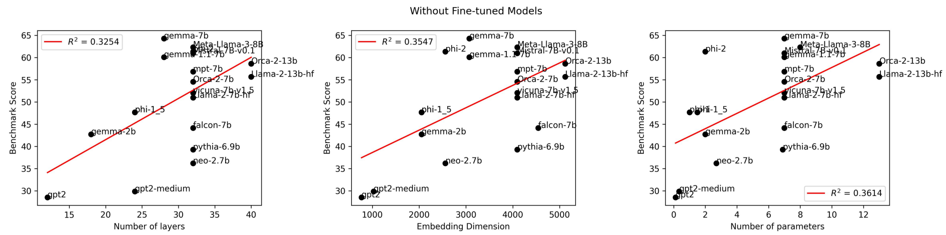


(b) Gemma 7B

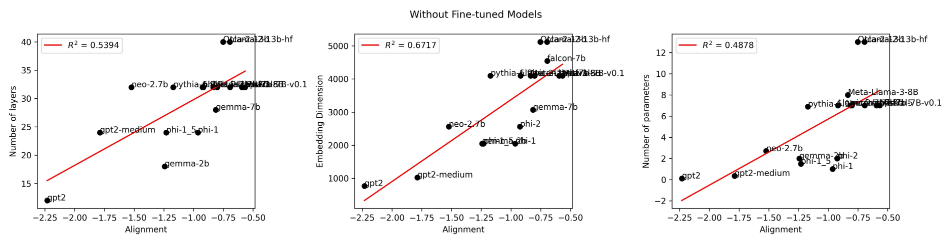


(c) GPT-2 XL

Figure 34: **Scaling of Singular Values of Residual Jacobians (Trained)**.

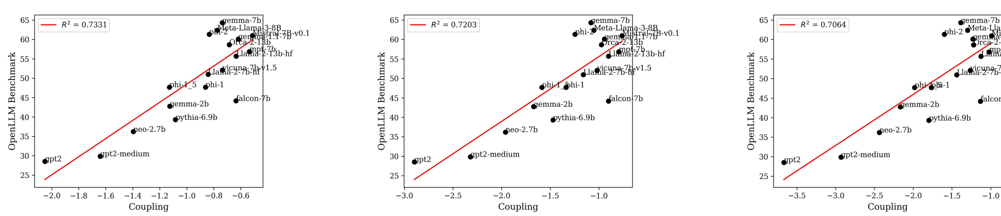


(a) Score vs. hyperparameters



(b) Hyperparameters vs. Alignment

Figure 35: **LLM number of layers, embedding dimension, and number of parameters, against score and Residual Jacobian Alignment**.



(a) $K = 25$ (b) $K = 50$ (c) $K = 75$

Figure 36: **Coupling plotted against benchmark score for varying number of singular vectors**.

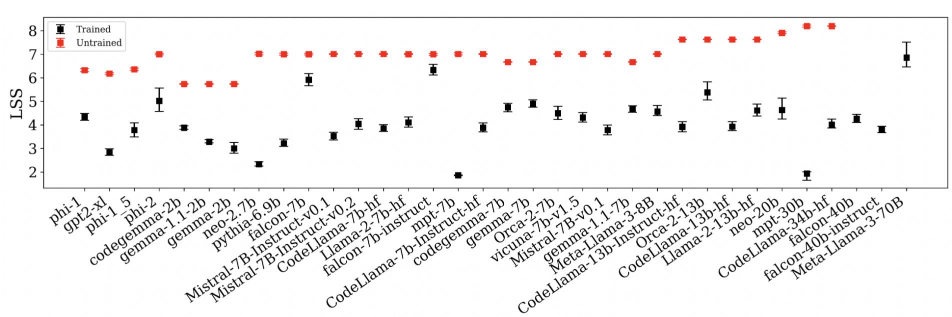
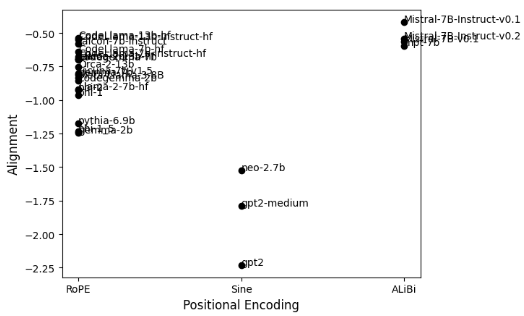
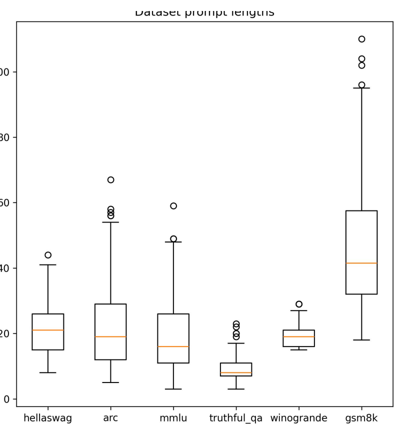


Figure 37: **LSS sorted by LLM parameters**.

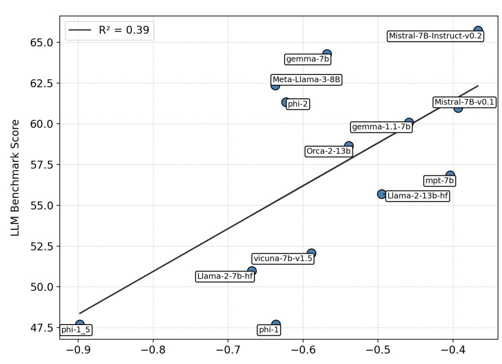


(a) Positional encoding and Alignment

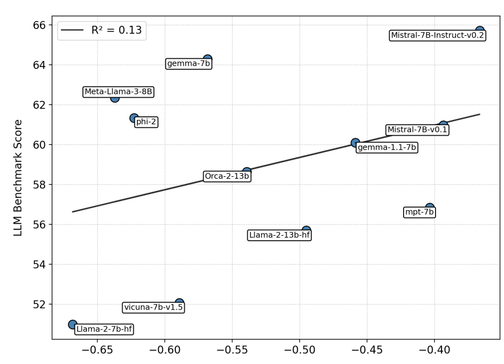


(b) Prompt lengths

Figure 38: **Other plots**.



(a)



(b)

Figure 39: **Restricted variants of Figure 1** a.