How Much New Geometry Does Another Training Run Add? Effective-Dimension Scaling Across Aligned Parameter Trajectories

Anonymous authors
Paper under double-blind review

Abstract

Neural network training evolves in a high-dimensional parameter space, yet independent runs often exhibit shared geometric structure. Prior work has identified such structure in both converged solutions and training trajectories. How the joint spectrum of complete parameter trajectories changes with the number of independent runs remains less well understood. We organize checkpoints from multiple runs into a joint trajectory tensor and measure the effective dimension of its singular-value spectrum after symmetry-aware alignment and temporal centering. We compare the observed spectra with the weak and strong matched-rotation nulls and repeat the analysis across multiple nested trajectory orderings. Across the evaluated architectures and tasks, effective dimension grows rapidly over the first few runs, followed by diminishing marginal growth over the measured range of up to 40 runs. The observed dimension remains far below the strong matched-rotation null and the algebraic rank bound. These results show that aligned parameter trajectories have a strongly concentrated, effectively low-rank joint spectrum over the observed run-count range.

1 Introduction

How Many New Dimensions Does Each Additional Training Run Add? Neural network training unfolds in a high-dimensional parameter space, yet independently trained models can share geometric structure, as shown by mode connectivity and symmetry-aware alignment (Garipov et al., 2018; Ainsworth et al., 2023). Prior work has also studied the spectral structure of final models (Kaushik et al., 2025a) and properties of training trajectories (Mao et al., 2024; Xu, 2026). How the joint spectrum of the complete trajectories $\mathcal{T} = \{\theta_i(t) : i = 1, \ldots, N;\ t \in [0, T]\}$ evolves with the number of independent runs, and whether the resulting spectral concentration survives matched controls for parameterization redundancy and estimator artifacts, have not been systematically characterized. We measure the effective dimension $D(S)$ for subsets of $N = |S|$ trajectories and examine the growth of its mean $\widehat{D}(N)$ as runs are added.

Diminishing Marginal Growth. Across the measured range of up to 40 runs, cross-trajectory effective dimension grows rapidly over the first few runs and then gradually levels off as its marginal increments decrease. It remains roughly an order of magnitude below the algebraic rank bound and below both matched-rotation nulls. To assess measurement artifacts, we compare each $\widehat{D}(N)$ curve with weak and strong matched-rotation nulls using the same matrix dimensions, SVD solver, and statistics. The weak null preserves within-trajectory temporal coupling, whereas the strong null disrupts it. Although concentrated spectral heads can arise in high-dimensional random walks without shared structure (Antognini & Sohl-Dickstein, 2018), the observed curves remain below both nulls, while the strong null continues to grow toward the algebraic rank bound. This contrast shows that finite matrix size and the effective-dimensionality estimator alone do not explain the observed spectral concentration.

Our main contributions are summarized as follows:

Run-Count Scaling of Trajectory Dimensionality. We characterize finite-range dimensional growth and its diminishing marginal increments across the evaluated architectures and tasks.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

A Controlled Measurement Framework. Our framework uses known parameter symmetries and temporal centering to isolate trajectory variation. Data-dependent permutations or rotations align equivalent coordinates, while Haar-random rotations serve as matched controls that preserve or disrupt within-trajectory temporal coupling.

A Supporting Trajectory Dataset. Publicly available trajectory collections with dense intermediate checkpoints are limited. We therefore assemble public and newly fine-tuned trajectories spanning multiple architectures, datasets, task settings, and model scales.

2 Related Work

Mode Connectivity and Weight-Space Alignment. Independently trained solutions can be connected by low-loss paths, including near-linear interpolations after accounting for parameter symmetries (Hochreiter & Schmidhuber, 1997; Choromanska et al., 2015; Freeman & Bruna, 2017; Garipov et al., 2018; Draxler et al., 2018; Frankle et al., 2020). Alignment also enables models to be matched, merged, or composed in weight space (Entezari et al., 2022; Ainsworth et al., 2023; Wortsman et al., 2022; Ilharco et al., 2023; Yadav et al., 2023; Yu et al., 2024; Yang et al., 2024). Transformer symmetries further include continuous rotations (Zhang et al., 2025). These studies primarily concern relationships among trained solutions.

Intrinsic Dimensionality. Intrinsic dimension has been estimated by optimizing in random affine subspaces (Li et al., 2018) and by measuring the update dimensions required for language model fine-tuning (Aghajanyan et al., 2021). Related work studies subspaces occupied by gradients, individual training trajectories, and parameter updates (Gur-Ari et al., 2018; Li et al., 2023; Song et al., 2025), reusable update subspaces (Kaushik et al., 2025b), and shared spectral subspaces across converged weights and adapters (Kaushik et al., 2025a).

Recurring Structures and Training Dynamics. Independent training reproduces Gabor-like early filters (Krizhevsky et al., 2012), recurring interpretable features (Olah et al., 2020; Chughtai et al., 2023), and common encodings and spectral directions (Guth & Ménard, 2024; Guth et al., 2024). Related studies examine shared subspaces among converged weights (Kaushik et al., 2025a), low-dimensional prediction trajectories for the same task (Mao et al., 2024), manifolds learned from parameter trajectories to guide training (Lu et al., 2023), transferred learning processes (Chijiwa, 2024), and the effective rank of individual Transformer parameter trajectories (Xu, 2026). We instead align collections of independent parameter trajectories, retain their temporal evolution, and measure how their joint spectrum changes as runs are added.

3 Measuring Joint Geometry Across Parameter Trajectories

We analyze the joint geometry of layer-wise parameter trajectories from independent runs for the same architecture and task (Figure 1). Checkpoints are tensorized, symmetry-aligned when required, temporally centered, and analyzed spectrally under two unfolding strategies. Repeating the analysis over nested trajectory subsets measures finite-range dimensional growth and marginal changes as runs are added.

3.1 Trajectory Representation and Preprocessing

Trajectory Representation. Let $\mathcal{R} = \{1, \dots, M\}$ denote the available independent training runs. For run $i \in \mathcal{R}$, let $W_i^{(\ell)}(t) \in \mathbb{R}^{d_{\text{out}}^{(\ell)} \times d_{\text{in}}^{(\ell)}}$ denote the weight matrix of layer ℓ at checkpoint t. All runs within an experimental setting follow the same training schedule and are sampled at T shared checkpoint indices. We use the checkpoint index as the common temporal coordinate and apply no temporal warping. Fully connected layers and the $Q/K/V/O$ projections in attention layers use their weight matrices directly. For convolutional layers, we consider two representations. In the output-mode representation, the spatial dimensions are flattened together with the input channels. In the input-mode representation, the spatial dimensions are flattened together with the output channels. Complete matrixization rules and shapes are provided in Appendix A.1.

For a selected trajectory subset $S = \{i_1, \ldots, i_N\} \subseteq \mathcal{R}$ with $N = |S|$, we construct the layer-wise trajectory tensor

$$\mathcal{X}_S^{(\ell)} \in \mathbb{R}^{d_{\text{out}}^{(\ell)} \times d_{\text{in}}^{(\ell)} \times T \times N}, \qquad \left(\mathcal{X}_S^{(\ell)}\right)_{a,b,t,n} = \left(W_{i_n}^{(\ell)}(t)\right)_{a,b}. \tag{1}$$

Its four modes represent output coordinates, input coordinates, training time, and independent training runs. For the input-mode representation of convolutional layers, the roles of the input and output channels are exchanged.

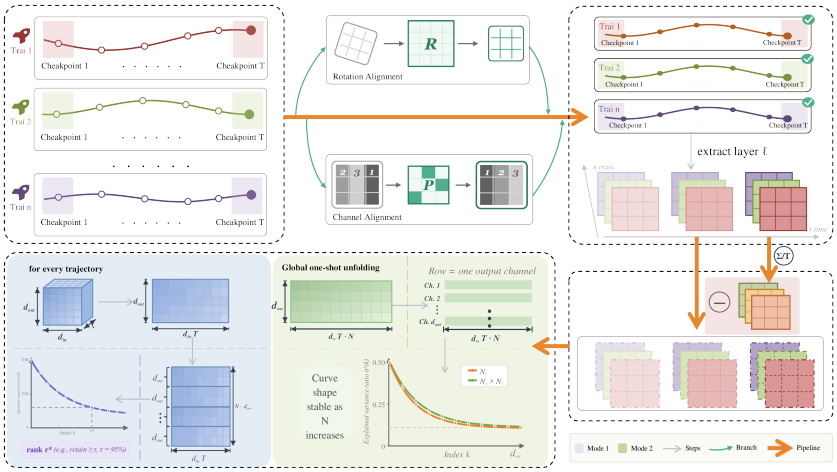


Figure 1: Overview of the cross-trajectory spectral analysis. Given a selected subset S of $N = |S|$ trajectories, layer-wise checkpoints are organized into trajectory tensors. We align the trajectories to remove function-preserving coordinate differences, after which they are temporally centered and analyzed spectrally under two unfolding strategies.

Symmetry-Aware Alignment. Function-preserving symmetries can place equivalent models in different parameter coordinates. For each trajectory i in a selected subset S, we estimate a time-invariant transformation $g_{i,S}^{\star}$ from its final checkpoint and apply it to the complete trajectory. For MLP, CNN, and Transformer FFN layers, we align with coupled channel permutations (Entezari et al., 2022; Ainsworth et al., 2023), while for self-attention we align with head-wise orthogonal rotations of the query-key and value-output pairs (Zhang et al., 2025). For each S, a single-pass reference strategy selects the trajectory closest to the centroid of the final checkpoints and independently aligns the remaining trajectories to it. The reference and transformations are recomputed for every subset, so the aligned quantities depend on S; we suppress this subscript for readability. The group actions, objectives, and solvers are detailed in Appendix A.2.

Temporal Centering. Let $\widetilde{W}_i^{(\ell)}(t)$ denote the aligned weight of layer ℓ at checkpoint t. To isolate temporal variation from static parameter offsets, which can dominate the spectrum when runs share a pretrained initialization, we subtract each trajectory's temporal mean before unfolding:

$$W_{i,\mathrm{c}}^{(\ell)}(t) = \widetilde{W}_i^{(\ell)}(t) - \frac{1}{T}\sum_{u=1}^{T} \widetilde{W}_i^{(\ell)}(u). \tag{2}$$

The centered tensor $\mathcal{X}_{\mathrm{c},S}^{(\ell)}$ collects these weights, preserving the original shape and all T checkpoints while giving each trajectory zero temporal mean.

3.2 Two Unfolding Strategies

We describe the output mode and omit the layer index ℓ for clarity. For fully connected layers and attention projections, the input mode is defined analogously by exchanging d_{out} and d_{in}; convolutional layers use the corresponding input-channel matrixization described in Appendix A.1.

Mode 1: Trajectory-Wise Unfolding and Stacking. For trajectory i, we flatten the input and time dimensions of the centered weights into a single feature axis:

$$X_i^{(\mathrm{out})} \in \mathbb{R}^{d_{\mathrm{out}} \times (d_{\mathrm{in}} T)}, \qquad \big(X_i^{(\mathrm{out})}\big)_{a,(b,t)} = \big(W_{i,\mathrm{c}}(t)\big)_{a,b}. \tag{3}$$

Each row represents the temporal evolution of one output coordinate, concatenated over all input coordinates and checkpoints.

For a trajectory subset $S = \{i_1, \ldots, i_N\}$ with $N = |S|$, we stack the trajectory matrices along the row axis:

$$M_{\mathrm{out}}^{(1)}(S) = \Big[\big(X_{i_1}^{(\mathrm{out})}\big)^\top \ \cdots \ \big(X_{i_N}^{(\mathrm{out})}\big)^\top\Big]^\top \in \mathbb{R}^{(N d_{\mathrm{out}}) \times (d_{\mathrm{in}} T)}. \tag{4}$$

The resulting rows correspond to trajectory–output-coordinate pairs. Symmetry alignment makes the coordinates comparable across runs.

Mode 1 retains all T checkpoints, so its matrix has $d_{\mathrm{in}} T$ columns. Temporal centering imposes one linear constraint per input coordinate, giving

$$\mathrm{rank}\big(M_{\mathrm{out}}^{(1)}(S)\big) \le \min\left(N d_{\mathrm{out}}, d_{\mathrm{in}}(T-1)\right). \tag{5}$$

The complete output- and input-mode constructions, Gram-matrix derivations, and stacking-order analysis are provided in Appendix A.3.

Mode 2: Global Unfolding Across Trajectories. Mode 2 unfolds the centered trajectory tensor jointly over the selected trajectory subset. For $S = \{i_1, \ldots, i_N\}$, the output-mode unfolding is

$$M_{\mathrm{out}}^{(2)}(S) = \Big[X_{i_1}^{(\mathrm{out})} \quad X_{i_2}^{(\mathrm{out})} \quad \cdots \quad X_{i_N}^{(\mathrm{out})}\Big] \in \mathbb{R}^{d_{\mathrm{out}} \times (N d_{\mathrm{in}} T)}. \tag{6}$$

Each row describes one aligned output coordinate across all input coordinates, checkpoints, and selected trajectories. Unlike Mode 1, whose rows are trajectory and output-coordinate pairs, Mode 2 retains a fixed coordinate space with rank bound $\mathrm{rank}\big(M_{\mathrm{out}}^{(2)}(S)\big) \le d_{\mathrm{out}}$. We therefore examine changes in the normalized spectrum to assess whether spectral concentration persists in this fixed coordinate space. The input-mode construction is obtained analogously. The full rank bound, complete constructions, Gram-matrix derivations, and trajectory-order analysis are provided in Appendix A.4.

For each trajectory subset, unfolding strategy, and parameter mode, we compute the singular values of the resulting matrix. Their squared values are normalized to obtain the spectral energy used in the next subsection.

3.3 Spectral Statistics and Run-Count Analysis

Spectral Statistics. For a trajectory subset S, let $M(S)$ denote the matrix produced by a specified layer, parameter mode, and unfolding strategy; we omit these indices below. Let $\sigma_{S,1} \ge \cdots \ge \sigma_{S,K(S)} > 0$ be its nonzero singular values. The normalized spectral energy is

$$p_{S,j} = \frac{\sigma_{S,j}^2}{\sum_{k=1}^{K(S)} \sigma_{S,k}^2}, \qquad j = 1, \ldots, K(S). \tag{7}$$

We use two effective-dimensionality statistics. For an energy threshold $\tau \in (0,1)$, the threshold dimension counts the directions needed to explain at least τ of the spectral energy. The entropy-based dimension measures how broadly energy is distributed over the full spectrum:

$$D_\tau(S) = \min\left\{k : \sum_{j=1}^{k} p_{S,j} \ge \tau\right\}, \quad D_{\mathrm{ent}}(S) = \exp\left(-\sum_{j=1}^{K(S)} p_{S,j} \log p_{S,j}\right). \tag{8}$$

We use $\tau = 0.95$ as the primary threshold and report additional thresholds as sensitivity analyses. We write $D(S)$ when the definition applies to either statistic and specify the statistic in each result.

Run-Count Analysis. To estimate how dimensionality changes with the number of trajectories, let π_b denote an ordering of the available runs $\mathcal{R} = \{1, \ldots, M\}$. We construct nested subsets

$$S_{b,N} = \{\pi_b(1), \ldots, \pi_b(N)\}, \qquad N = 1, \ldots, M. \tag{9}$$

For the primary attention analyses, let d index the dataset, $p \in \{Q, K, V, O\}$ the projection matrix, $a \in \{\text{in}, \text{out}\}$ the retained parameter axis, and $b = 1, \ldots, B$ the nested-subset sequence. For statistic m, the reported mean $\widehat{D}_d^{(m)}(N)$ and standard deviation $s_d^{(m)}(N)$ are computed over b, p, and a from $D_{d,p,a}^{(m)}(S_{b,N})$. The standard deviation describes their joint dispersion, not a confidence interval. Here, $m = 0.95$ denotes the primary energy-threshold dimension, $m = \text{ent}$ the entropy-based dimension, and other values of τ the threshold-sensitivity analyses. When the dataset is identified in a table or figure, we abbreviate $\widehat{D}_d^{(m)}(N)$ as $\widehat{D}^{(m)}(N)$.

Let $\mathcal{N} = \{N_1, \ldots, N_K\}$, where $N_1 < \cdots < N_K$, denote the evaluated trajectory counts. For consecutive counts, the average effective-dimension change per added trajectory is

$$\overline{\Delta D}_{b,k} = \frac{D(S_{b,N_k}) - D(S_{b,N_{k-1}})}{N_k - N_{k-1}}, \qquad k = 2, \ldots, K. \tag{10}$$

Because the alignment reference and transformations are recomputed for each subset, this quantity measures the average change between two subset-specific analyses rather than the isolated effect of a single added trajectory. We summarize these changes without assuming monotonicity.

4 Experiments

4.1 Setup

We analyze CNN and attention-based trajectories from the public NiNo checkpoint dataset and from our DeBERTa-v3 fine-tuning runs. Table 4 in Appendix A.6 summarizes the model architectures, trajectory counts, and checkpoint resolution. Each model-dataset pair is analyzed separately, except in the cross-dataset mixing experiment. Trajectories are never pooled across model sizes.

NiNo Trajectories. We use four subsets of the publicly available NiNo checkpoint dataset (Knyazev et al., 2025). The CNN subsets contain models trained on FashionMNIST (FM-16) and CIFAR10 (C10-16), each with three convolutional layers of 16, 32, and 32 channels. The two GPT2-style Transformer subsets are trained on LM1B. The LM1B-3-24 models contain 3 layers, a hidden dimension of 24, and 3 attention heads, while the LM1B-2-32 models contain 2 layers, a hidden dimension of 32, and 2 attention heads.

DeBERTa-v3 Trajectories. We independently fine-tune DeBERTa-v3-`xsmall`, DeBERTa-v3-`small`, and DeBERTa-v3-`base` for emotion classification (He et al., 2023). The `xsmall` and `small` variants cover eight datasets: CrowdFlower, TEC, Tales-Emotion, ISEAR, EmoInt, ElectoralTweets, Grounded Emotions, and Emotion-Cause (CrowdFlower, 2016; Mohammad, 2012; Alm et al., 2005; Scherer & Wallbott, 1994; Mohammad & Bravo-Marquez, 2017; Mohammad et al., 2015; Liu et al., 2017; Ghazi et al., 2015). The `base` variant is evaluated on Grounded Emotions and Emotion-Cause as a supplementary cross-scale comparison. We use the unified collection of Bostan & Klinger (2018), which harmonizes the dataset formats, label taxonomy, and preprocessing pipeline. The fine-tuning hyperparameters follow Jin et al. (2023), with the learning rate and batch size adjusted by model size. For each combination of model and dataset, 40 runs start from the same pretrained checkpoint and use different random seeds.

Default Analysis Protocol. Unless otherwise stated, the main experiments first align trajectories using the symmetries of each architecture and then subtract each trajectory's temporal mean. We apply Mode 1 at the first analyzed layer, considering the input and output parameter axes separately. The primary statistic is $D_{0.95}$; D_{ent} and Mode 2 provide complementary analyses.

For each model-dataset pair, we evaluate $N \in \{1, 2, 3, 4, 5, 10, 20, 30, 40\}$ using a fixed collection of $B = 16$ trajectory orderings: one prespecified ordering and 15 randomly generated permutations. These orderings are generated once and reused across all analyses that follow this protocol, including different layers, projections, and retained axes. Each ordering defines a nested sequence of subsets by selecting its first N trajectories.

For attention analyses, we evaluate the Q, K, V, and O projections separately along each retained axis, yielding eight projection-axis configurations. Unless otherwise stated, the per-dataset Mode-1 results report the mean and standard deviation across the $16 \times 8 = 128$ estimates at each N. Experiment-specific departures from this protocol are stated locally. Additional ablations examine

temporal centering, symmetry alignment, checkpoint sampling density, and layer-wise consistency (Appendix B).

4.2 Finite-Range Scaling of Cross-Trajectory Effective Dimension

Scaling with the Number of Trajectories. Table 1 reports both effective-dimension statistics and their standard deviations for DeBERTa-v3-`xsmall` across the eight emotion datasets. Both show rapid initial growth followed by smaller increases as more trajectories are included.

Table 1: Finite-range effective-dimensionality estimates for DeBERTa-v3-`xsmall`. Entries report means $\pm$ standard deviations under the default aggregation protocol in Section 4.1, rounded to integers. CF, Tales, Electoral, Grounded, and EmoCause denote CrowdFlower, Tales-Emotion, ElectoralTweets, Grounded Emotions, and Emotion-Cause, respectively.

# Traj.	Metric	CF	TEC	Tales	ISEAR	EmoInt	Electoral	Grounded	EmoCause
1	$\widehat{D}_{0.95}$	218 ± 16	227 ± 19	203 ± 17	203 ± 17	209 ± 19	168 ± 23	207 ± 9	155 ± 16
	$\widehat{D}_{\mathrm{ent}}$	132 ± 21	145 ± 27	120 ± 23	117 ± 21	103 ± 25	79 ± 32	114 ± 16	59 ± 12
2	$\widehat{D}_{0.95}$	293 ± 22	302 ± 25	253 ± 20	280 ± 25	299 ± 29	215 ± 31	306 ± 15	222 ± 26
	$\widehat{D}_{\mathrm{ent}}$	166 ± 26	180 ± 34	141 ± 26	150 ± 28	134 ± 33	96 ± 39	161 ± 14	79 ± 16
3	$\widehat{D}_{0.95}$	337 ± 24	342 ± 28	281 ± 22	328 ± 28	352 ± 35	236 ± 33	362 ± 20	258 ± 31
	$\widehat{D}_{\mathrm{ent}}$	183 ± 29	196 ± 38	151 ± 29	169 ± 31	150 ± 37	103 ± 42	187 ± 18	88 ± 18
4	$\widehat{D}_{0.95}$	363 ± 24	368 ± 30	304 ± 21	359 ± 31	387 ± 41	246 ± 34	400 ± 21	283 ± 32
	$\widehat{D}_{\mathrm{ent}}$	193 ± 30	205 ± 40	160 ± 30	180 ± 33	159 ± 41	106 ± 43	205 ± 18	93 ± 19
5	$\widehat{D}_{0.95}$	386 ± 25	387 ± 31	317 ± 22	382 ± 33	414 ± 43	253 ± 34	427 ± 20	302 ± 33
	$\widehat{D}_{\mathrm{ent}}$	201 ± 32	212 ± 41	164 ± 30	188 ± 35	166 ± 43	108 ± 44	217 ± 20	97 ± 20
10	$\widehat{D}_{0.95}$	435 ± 26	435 ± 36	348 ± 24	444 ± 39	490 ± 52	280 ± 35	509 ± 23	354 ± 37
	$\widehat{D}_{\mathrm{ent}}$	217 ± 33	228 ± 45	174 ± 32	209 ± 40	186 ± 48	116 ± 46	251 ± 26	111 ± 22
20	$\widehat{D}_{0.95}$	467 ± 28	470 ± 39	369 ± 25	486 ± 43	545 ± 57	297 ± 35	578 ± 23	388 ± 39
	$\widehat{D}_{\mathrm{ent}}$	226 ± 35	238 ± 47	180 ± 33	220 ± 42	199 ± 51	120 ± 48	275 ± 33	118 ± 24
30	$\widehat{D}_{0.95}$	483 ± 28	486 ± 40	379 ± 25	504 ± 45	569 ± 59	305 ± 35	612 ± 25	403 ± 40
	$\widehat{D}_{\mathrm{ent}}$	231 ± 35	242 ± 47	182 ± 34	225 ± 43	203 ± 52	122 ± 48	286 ± 38	122 ± 24
40	$\widehat{D}_{0.95}$	490 ± 29	492 ± 41	383 ± 26	512 ± 46	583 ± 60	308 ± 35	628 ± 31	414 ± 40
	$\widehat{D}_{\mathrm{ent}}$	232 ± 35	244 ± 48	183 ± 34	226 ± 43	206 ± 52	123 ± 49	289 ± 40	124 ± 25

Interval-Averaged Marginal Changes. To quantify this change, we compute the interval-averaged increase between consecutive trajectory counts according to Equation 10. As shown in Figure 2, under both statistics, the interval-averaged increase decreases across all eight datasets as more trajectories are added. Over the final interval from $N = 30$ to $N = 40$, $\overline{\Delta D}^{(0.95)}$ ranges from 0.38 to 1.66 dimensions per added trajectory, while $\overline{\Delta D}^{(\mathrm{ent})}$ ranges from 0.08 to 0.36. The effective dimension therefore exhibits diminishing marginal growth and remains in a low-growth regime over the largest observed trajectory counts.

Further analyses of the energy-threshold and entropy-based dimensions, including threshold sensitivity and complete dataset-wise scaling curves, are provided in Appendix B.5.

Matched-Null Controls. To test whether the observed diminishing marginal growth could arise from the finite matrix size, SVD solver, or choice of statistic, we construct two matched null models on DeBERTa-v3-`xsmall`. The null transformations are applied after alignment and temporal centering. Both controls retain the matrix dimensions, SVD solver, and statistics of the observed analysis. Using the QR construction of Mezzadri (2007), the nulls draw independent Haar-random matrices from $O(d)$. For each nested subset $S_{b,N}$, we independently sample one realization of each null. These rotations preserve each checkpoint's singular values and norm.

The weak null rotates each trajectory with an independently sampled pair of orthogonal matrices shared across all its checkpoints. This preserves within-trajectory temporal structure and the temporal-centering constraint while randomizing cross-trajectory coordinate correspondence. The strong null rotates each trajectory and checkpoint pair independently. This additionally disrupts within-trajectory temporal coherence and removes the temporal zero-sum constraint. For the square attention projections used here, the strong-null Mode-1 rank bound for both retained axes is $\min(Nd_{\mathrm{out}}, d_{\mathrm{in}}T)$.

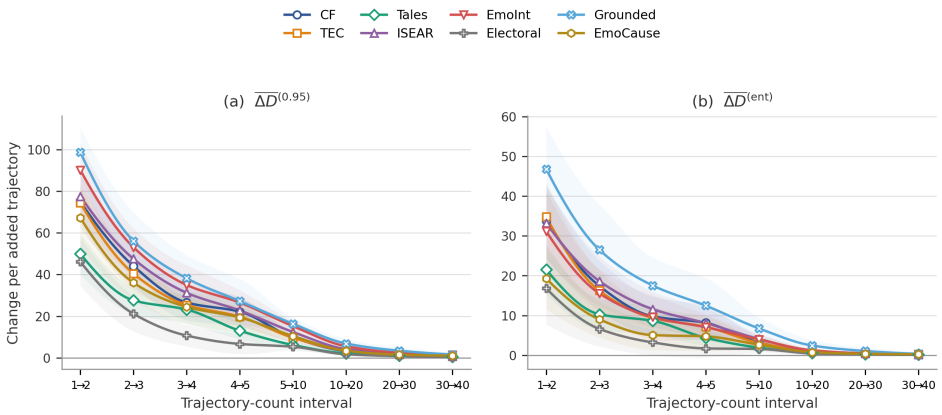


Figure 2: Interval-averaged effective-dimension changes per added trajectory across the eight datasets. The left panel reports $\overline{\Delta D}^{(0.95)}$, and the right panel reports $\overline{\Delta D}^{(\mathrm{ent})}$. For each dataset, lines show mean interval-averaged changes, and shaded bands show the corresponding standard deviations.

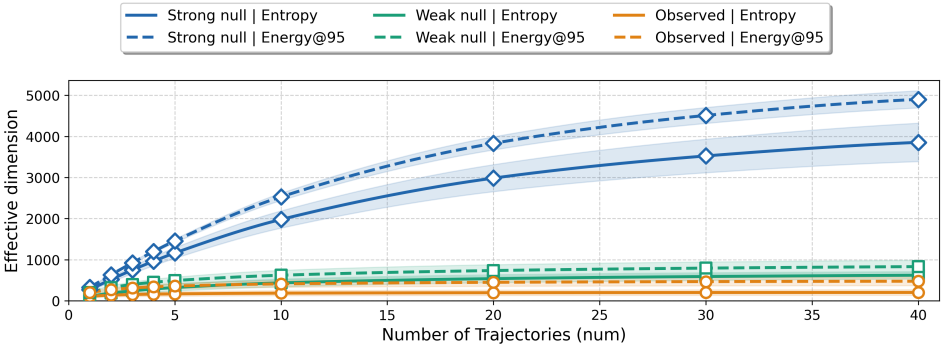


Figure 3: Effective dimension of observed trajectories and matched-rotation nulls on DeBERTa-v3-`xsmall`. Solid and dashed lines show pooled means of $D_{\mathrm{ent}}(S)$ and $D_{0.95}(S)$, respectively. All three conditions use the same trajectory subsets, with one independent realization of each null per ordering and trajectory count. Results additionally pool the eight datasets; standard deviations for the nulls also include variation from random rotations.

Figure 3 shows that at $N = 1$, the weak null agrees with the observed result up to numerical precision, as a fixed orthogonal transformation leaves the joint spectrum unchanged. For $N > 1$, the weak null produces larger effective dimensions than the observed trajectories. At $N = 40$, its mean $D_{0.95}$ and D_{ent} are approximately 1.75 and 3.06 times the observed values, respectively. Randomizing cross-trajectory coordinate correspondence therefore increases the measured dimension, consistent with the alignment ablation in Appendix B.2. The weak null still exhibits diminishing marginal growth at a higher effective dimension. The strong null grows substantially faster; at $N = 40$, its mean $D_{0.95}$ reaches 4901.02, or 63.8% of the rank bound of 7680.

From $N = 30$ to $N = 40$, the mean change per added trajectory in $D_{0.95}$ is 0.89 for the observed trajectories, compared with 3.64 under the weak null and 39.14 under the strong null. For D_{ent}, the corresponding changes are 0.19, 3.11, and 33.09. This contrast shows that the matrix dimensions, SVD solver, and statistics alone do not impose the low late-range growth observed in the original trajectories.

4.3 Mode-2 Spectral Stability in the Fixed coordinate Space

Mode 2 examines whether the normalized spectral energy remains concentrated as trajectories are added. We compare the sorted energy profiles across trajectory counts and against the matched nulls. Figure 4 shows that the observed profiles retain similar shapes as N increases, particularly after the first few trajectories are included. The weak null shows a clearer reduction in leading-direction energy as N grows. This change is strongest under the strong null, where the spectrum becomes progressively flatter and the energy is distributed across more coordinate directions. The contrast shows that the concentration and stability of the observed spectrum cannot be explained by the fixed coordinate-space rank bound alone. These results support persistent spectral concentration in the aligned coordinate space, but do not establish that the leading subspaces are identical across trajectory subsets.

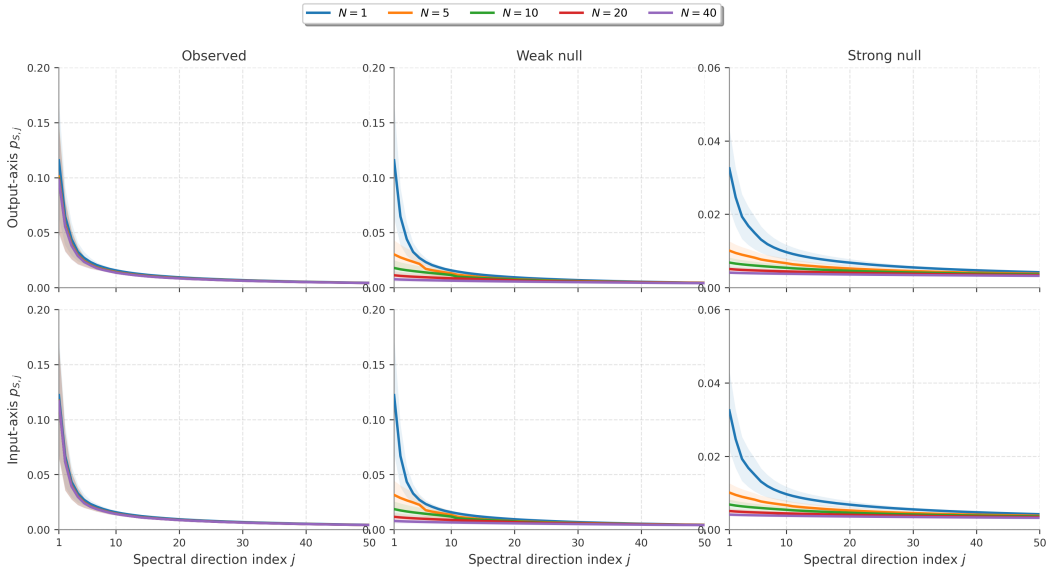


Figure 4: Mode-2 normalized spectral energy profiles for observed trajectories, weak null, and strong null (columns); rows correspond to the output and input axes. Each panel shows $N \in \{1, 5, 10, 20, 40\}$ and the leading 50 directions of spectra normalized over all 384 directions. Within each row, curves and bands show pointwise means and standard deviations across trajectory orderings, the eight datasets, and attention projections. Panels use separate linear vertical scales; tick labels report energy fractions.

4.4 Generalization Across Datasets and Model Scales

Cross-Dataset Mixing at Fixed Trajectory Budgets. For each $k \in \{1, \ldots, 8\}$, we sample $R_k = \min\{\binom{8}{k}, 28\}$ distinct combinations of k datasets. Within each combination, trajectories are allocated as evenly as possible across the k datasets for $N \in \{30, 40\}$. The ordered $N = 30$ trajectory list is a strict prefix of the ordered $N = 40$ list. At $k = 1$, the eight single-dataset combinations use fixed trajectory orders. No trajectory-order resampling is performed.

Figure 5 shows the results. Effective dimension increases as more datasets are mixed, indicating a less concentrated joint spectrum and supporting greater heterogeneity in training dynamics. The paired change from $N = 30$ to $N = 40$ remains small across k. Thus, dataset diversity raises the overall dimensional level while growth over this late range remains limited.

Consistency Across Model Scales. We compare DeBERTa-v3-`xsmall`, `small`, and `base` on Grounded Emotions and Emotion-Cause, the two datasets available at all three scales. Each model-dataset pair contributes 40 trajectories in a fixed order. We use the trajectory counts in Section 4.1 and do not resample the order.

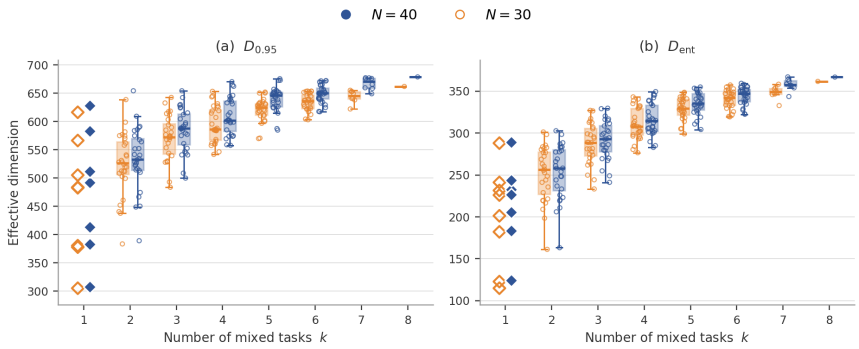


Figure 5: Effective dimension of cross-dataset mixtures as the number of included datasets k increases. The left panel reports $D_{0.95}$, and the right reports D_{ent}. Hollow and filled markers denote $N = 30$ and $N = 40$. Points represent dataset combinations, and boxes summarize their distributions. Each point averages the eight projection-axis configurations, without trajectory-order resampling.

Figure 6 shows the results. The absolute effective dimension depends on both model size and dataset. All three model sizes nevertheless retain the same qualitative finite-range pattern: rapid growth at small N followed by smaller marginal changes. The observed behavior is therefore not limited to a single model scale.

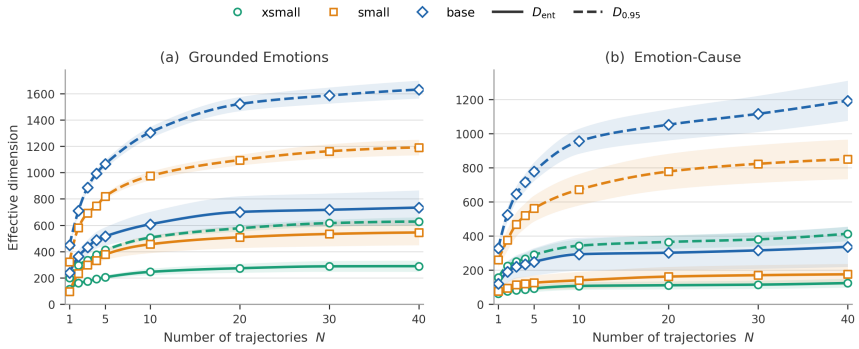


Figure 6: Effective-dimension curves across DeBERTa-v3 model sizes. The left and right panels report results on Grounded Emotions and Emotion-Cause, respectively. Colors distinguish `xsmall`, `small`, and `base`; solid and dashed lines denote $D_{\text{ent}}(N)$ and $D_{0.95}(N)$. For each model-dataset pair, a single fixed trajectory ordering is used. Lines and shaded bands show means and standard deviations across the eight projection-axis configurations.

5 Conclusion

Across the evaluated emotion datasets and DeBERTa-v3 model sizes, aligned training trajectories exhibit rapid initial growth in effective dimension followed by diminishing marginal growth over the measured range of up to 40 runs. Mixing datasets raises the dimensional level while retaining small late-range changes. Mode 2 normalized spectral energy profiles remain concentrated, and matched-rotation nulls yield more dispersed spectra under the same matrix dimensions and spectral estimators.

These results characterize finite-range spectral concentration rather than an asymptotic limit. Their scope is limited to the evaluated architectures and known permutation and rotation symmetries. Explaining the shared spectral structure and its relationship to functional diversity remains an open question.

AI USE STATEMENT

In this work, we used generative AI tools to aid or polish writing, and for retrieval and discovery of related work (e.g., AI-assisted literature search and reading during the early survey stage). We have not used generative AI tools for research ideation or execution, for drafting sections of the paper, for generating synthetic datasets, or for proving mathematical claims, and no other required disclosure tasks apply to this work. Additionally, we have not used generative AI tools for any other tasks with recommended disclosure. We have reviewed all AI-assisted work. All AI-assisted writing was limited to language polishing and has been reviewed, verified, and edited by the authors; all AI-assisted literature search results were read and understood in detail by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The main text specifies the trajectory construction, alignment, temporal centering, unfolding strategies, spectral estimators, and statistical aggregation protocol. The appendix provides the corresponding implementation details, tensor shapes, symmetry actions, and numerical procedures. The experimental section reports the model architectures, datasets, checkpoint sampling schedules, trajectory counts, and evaluation settings. The complete implementation, together with the experiment configurations and environment specification, is publicly available at `https://anonymous.4open.science/r/REPO-D3D8`.

REFERENCES

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7319–7328, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.568. URL `https://aclanthology.org/2021.acl-long.568/`.

Samuel Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=CQsmMYmlP5T`.

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. Emotions from text: Machine learning for text-based emotion prediction. In Raymond Mooney, Chris Brew, Lee-Feng Chien, and Katrin Kirchhoff (eds.), *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pp. 579–586, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics. URL `https://aclanthology.org/H05-1073/`.

Joseph Antognini and Jascha Sohl-Dickstein. PCA of high dimensional random walks with comparison to neural network training. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper_files/paper/2018/file/7a576629fef88f3e636afd33b09e8289-Paper.pdf`.

Laura-Ana-Maria Bostan and Roman Klinger. An analysis of annotated corpora for emotion classification in text. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle (eds.), *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2104–2119, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL `https://aclanthology.org/C18-1179/`.

Daiki Chijiwa. Transferring learning trajectories of neural networks. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=bWNJFD1l8M`.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gerard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In Guy Lebanon and S. V. N. Vishwanathan (eds.), *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pp. 192–204, San Diego, California, USA, 09–12 May 2015. PMLR. URL https://proceedings.mlr.press/v38/choromanska15.html.

Bilal Chughtai, Lawrence Chan, and Neel Nanda. A toy model of universality: reverse engineering how networks learn group operations. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

CrowdFlower. The emotion in text dataset. CrowdFlower Data for Everyone Library, 2016. URL https://github.com/sarnthil/unify-emotion-datasets/blob/master/datasets/README.md. Original dataset URL: http://www.crowdflower.com/wp-content/uploads/2016/07/text_emotion.csv (no longer accessible). Documentation and dataset access via the unified collection of Bostan and Klinger (2018).

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1309–1318. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/draxler18a.html.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=dNigytemkL.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3259–3269. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/frankle20a.html.

C. Daniel Freeman and Joan Bruna. Topology and geometry of half-rectified network optimization. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Bk0FWVcgx.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of DNNs. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/be3087e74e9100d4bc4c6268cdbe8456-Paper.pdf.

Diman Ghazi, Diana Inkpen, and Stan Szpakowicz. Detecting emotion stimuli in emotion-bearing sentences. In *Conference on Intelligent Text Processing and Computational Linguistics*, 2015. URL https://api.semanticscholar.org/CorpusID:18306963.

Guy Gur-Ari, Daniel A. Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *ArXiv*, abs/1812.04754, 2018. URL https://api.semanticscholar.org/CorpusID:54480858.

Florentin Guth and Brice Ménard. On the universality of neural encodings in CNNs. In *ICLR 2024 Workshop on Representational Alignment*, 2024. URL https://openreview.net/forum?id=ofEBFOrITI.

Florentin Guth, Brice Ménard, Gaspar Rochette, and Stéphane Mallat. A rainbow in deep network black boxes. *Journal of Machine Learning Research*, 25(350):1–59, 2024. URL http://jmlr.org/papers/v25/23-1573.html.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sE7-XhLxHA.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997. doi: 10.1162/neco.1997.9.1.1.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FCnohuR6AnM.

Prakhar Kaushik, Shravan Chaudhari, Ankit Vaidya, Rama Chellappa, and Alan Yuille. The universal weight subspace hypothesis, 2025a. URL https://arxiv.org/abs/2512.05117.

Prakhar Kaushik, Ankit Vaidya, Shravan Chaudhari, and Alan Yuille. EigenLoRAx: Recycling adapters to find principal subspaces for resource-efficient adaptation and inference. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 649–659, 2025b. doi: 10.1109/CVPRW67362.2025.00070.

Boris Knyazev, Abhinav Moudgil, Guillaume Lajoie, Eugene Belilovsky, and Simon Lacoste-Julien. Accelerating training with neuron interaction and nowcasting networks. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 29569–29588, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/48a4f41ff000b121eabe7b2491b24aef-Paper-Conference.pdf.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018.

Tao Li, Lei Tan, Zhehao Huang, Qinghua Tao, Yipeng Liu, and Xiaolin Huang. Low dimensional trajectory hypothesis is true: DNNs can be trained in tiny subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3411–3420, 2023. doi: 10.1109/TPAMI.2022.3178101.

Vicki Liu, Carmen Banea, and Rada Mihalcea. Grounded emotions. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pp. 477–483, Los Alamitos, CA, USA, October 2017. IEEE Computer Society. doi: 10.1109/ACII.2017.8273642. URL https://doi.ieeecomputersociety.org/10.1109/ACII.2017.8273642.

Ke Lu, Xiaotong He, Ze Qin, Xinyao Li, and Zhekai Du. Exploring low-dimensional manifolds of deep neural network parameters for improved model optimization. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, pp. 1667–1676, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701245. doi: 10.1145/3583780.3614873. URL https://doi.org/10.1145/3583780.3614873.

Jialin Mao, Itay Griniasty, Han Kheng Teoh, Rahul Ramesh, Rubing Yang, Mark K. Transtrum, James P. Sethna, and Pratik Chaudhari. The training process of many deep networks explores the same low-dimensional manifold. *Proceedings of the National Academy of Sciences of the United States of America*, 121(12):e2310002121, 2024. doi: 10.1073/pnas.2310002121. URL https://doi.org/10.1073/pnas.2310002121.

Francesco Mezzadri. How to generate random matrices from the classical compact groups. *Notices of the American Mathematical Society*, 54(5):592 – 604, May 2007. ISSN 0002-9920.

Saif Mohammad. #Emotional tweets. In ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pp. 246–255, 2012.

Saif Mohammad and Felipe Bravo-Marquez. Emotion intensities in tweets. In Nancy Ide, Aurélie Herbelot, and Lluís Màrquez (eds.), *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pp. 65–77, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-1007. URL https://aclanthology.org/S17-1007/.

Saif M. Mohammad, Xiaodan Zhu, Svetlana Kiritchenko, and Joel Martin. Sentiment, emotion, purpose, and style in electoral tweets. *Information Processing & Management*, 51(4):480–499, 2015. ISSN 0306-4573. doi: 10.1016/j.ipm.2014.09.003. URL https://www.sciencedirect.com/science/article/pii/S0306457314000880.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. URL https://distill.pub/2020/circuits/zoom-in.

Klaus R. Scherer and Harald G. Wallbott. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of Personality and Social Psychology*, 66(2):310–328, 1994. doi: 10.1037//0022-3514.66.2.310.

Minhak Song, Kwangjun Ahn, and Chulhee Yun. Does SGD really happen in tiny subspaces? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=v6iLQBoIJw.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Yongzhong Xu. Low-dimensional execution manifolds in transformer learning dynamics: Evidence from modular arithmetic tasks. *ArXiv*, abs/2602.10496, 2026. URL https://api.semanticscholar.org/CorpusID:285469930.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-Merging: Resolving interference when merging models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 7093–7115. Curran Associates, Inc., 2023. doi: 10.52202/075280-0310. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1644c9af28ab7916874f6fd6228a9bcf-Paper-Conference.pdf.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. AdaMerging: Adaptive model merging for multi-task learning. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 22743–22763, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/62868cc2fc1eb5cdf321d05b4b88510c-Paper-Conference.pdf.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are Super Mario: Absorbing abilities from homologous models as a free lunch. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57755–57775. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/yu24p.html.

Binchi Zhang, Zaiyi Zheng, Zhengzhang Chen, and Jundong Li. Beyond the permutation symmetry of transformers: The role of rotation for model fusion. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=wBJIO15pBV.

A METHOD DETAILS

A.1 TRAJECTORY CONSTRUCTION AND TENSORIZATION

The analysis includes weight matrices from fully connected layers, MLP and Transformer FFN layers, self-attention projections, and convolutional layers. Bias vectors, LayerNorm parameters, embedding matrices, and classifier heads are excluded.

Fully connected and FFN weights retain their native matrix form:

$$W_i^{(\ell)}(t) \in \mathbb{R}^{d_{\text{out}}^{(\ell)} \times d_{\text{in}}^{(\ell)}}. \tag{11}$$

The query, key, value, and output projections of self-attention are represented in the same form. Each projection is analyzed as a complete matrix rather than being divided into individual attention heads. When the query, key, and value projections are stored as a fused parameter, we first split them into the corresponding Q, K, and V matrices. The four attention projections are then processed separately.

For a convolutional layer, let

$$\mathcal{W}_i^{(\ell)}(t) \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times k_h \times k_w} \tag{12}$$

denote its kernel tensor. For the output mode, we retain the output-channel axis and flatten the input channel and spatial axes:

$$\left(W_{i,\text{out}}^{(\ell)}(t)\right)_{a,(b,u,v)} = \left(\mathcal{W}_i^{(\ell)}(t)\right)_{a,b,u,v}. \tag{13}$$

This gives

$$W_{i,\text{out}}^{(\ell)}(t) \in \mathbb{R}^{C_{\text{out}} \times (C_{\text{in}} k_h k_w)}. \tag{14}$$

For the input mode, we retain the input-channel axis and flatten the output channel and spatial axes:

$$\left(W_{i,\text{in}}^{(\ell)}(t)\right)_{b,(a,u,v)} = \left(\mathcal{W}_i^{(\ell)}(t)\right)_{a,b,u,v}. \tag{15}$$

The resulting matrix has shape

$$W_{i,\text{in}}^{(\ell)}(t) \in \mathbb{R}^{C_{\text{in}} \times (C_{\text{out}} k_h k_w)}. \tag{16}$$

Both constructions preserve the kernel values and change only their matrix indexing.

After matrixization, each selected parameter is represented as

$$W_i^{(\ell)}(t) \in \mathbb{R}^{d_{\text{out}}^{(\ell)} \times d_{\text{in}}^{(\ell)}}. \tag{17}$$

For a trajectory subset $S = \{i_1, \ldots, i_N\}$ with $N = |S|$, the layer-wise trajectory tensor is

$$\mathcal{X}_S^{(\ell)} \in \mathbb{R}^{d_{\text{out}}^{(\ell)} \times d_{\text{in}}^{(\ell)} \times T \times N}, \qquad \left(\mathcal{X}_S^{(\ell)}\right)_{a,b,t,n} = \left(W_{i_n}^{(\ell)}(t)\right)_{a,b}. \tag{18}$$

The first two axes contain the matrixized parameter coordinates. The third axis preserves checkpoint order. The fourth axis indexes the selected trajectories. Symmetry alignment and temporal centering do not change this shape.

A.2 ARCHITECTURE-SPECIFIC ALIGNMENT

We use architecture-specific symmetries to match each selected trajectory to a common parameter coordinate system. MLP, CNN, and Transformer FFN layers are matched through channel permutations. Self-attention layers are matched through head-wise orthogonal transformations. The pairwise matching procedures follow Ainsworth et al. (2023) and Zhang et al. (2025), respectively.

A.2.1 PERMUTATION ALIGNMENT

Let $W_i^{(\ell)}(T) \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ and $b_i^{(\ell)}(T) \in \mathbb{R}^{d_\ell}$ denote the endpoint parameters of layer ℓ from trajectory i. For permutation matrices $P_i^{(\ell)} \in \Pi_{d_\ell}$, the function-preserving group action is

$$W_i^{(\ell)}(T) \mapsto P_i^{(\ell)} W_i^{(\ell)}(T) \left(P_i^{(\ell-1)}\right)^\top, \qquad b_i^{(\ell)}(T) \mapsto P_i^{(\ell)} b_i^{(\ell)}(T), \tag{19}$$

where Π_d denotes the set of $d \times d$ permutation matrices. The same hidden-unit permutation acts on the output coordinates of one layer and the input coordinates of the following layer. For convolutional networks, the corresponding transformation acts on channel axes.

Given a source endpoint $\theta_i(T)$ and an anchor endpoint $\theta_r(T)$, we use the weight-matching objective

$$\begin{aligned} \min_{\{P_i^{(\ell)}\}} \sum_\ell \Big[\Big\| P_i^{(\ell)} W_i^{(\ell)}(T) \left(P_i^{(\ell-1)}\right)^\top - W_r^{(\ell)}(T) \Big\|_F^2 \\ + \Big\| P_i^{(\ell)} b_i^{(\ell)}(T) - b_r^{(\ell)}(T) \Big\|_2^2 \Big]. \end{aligned} \tag{20}$$

When present, bias parameters are included in the matching objective but excluded from the subsequent spectral analysis. The bias term above is omitted for layers without bias.

We solve this coupled objective using the weight-matching procedure of Ainsworth et al. (2023). The permutation variables are updated through coordinate descent. Each update reduces to a linear assignment problem and is solved with the Hungarian algorithm. For Transformer FFNs, the hidden dimensions of different blocks are not coupled. We therefore solve for the permutation of each FFN block independently, following the implementation used by Zhang et al. (2025).

A.2.2 ATTENTION ROTATION ALIGNMENT

For self-attention, we use the closed-form rotation matching procedure of Zhang et al. (2025). Consider head h with head dimension d_h. Let

$$R_{qk,i}^h, R_{vo,i}^h \in O(d_h)$$

denote the orthogonal transformations for the query-key and value-output pairs. Using column-vector bias notation, their function-preserving actions are

$$\begin{aligned} W_{Q,i}^h &\mapsto \left(R_{qk,i}^h\right)^\top W_{Q,i}^h, \quad b_{Q,i}^h \mapsto \left(R_{qk,i}^h\right)^\top b_{Q,i}^h, \\ W_{K,i}^h &\mapsto \left(R_{qk,i}^h\right)^\top W_{K,i}^h, \quad b_{K,i}^h \mapsto \left(R_{qk,i}^h\right)^\top b_{K,i}^h, \\ W_{V,i}^h &\mapsto \left(R_{vo,i}^h\right)^\top W_{V,i}^h, \quad b_{V,i}^h \mapsto \left(R_{vo,i}^h\right)^\top b_{V,i}^h, \\ W_{O,i}^h &\mapsto W_{O,i}^h R_{vo,i}^h, \qquad\quad b_{O,i} \mapsto b_{O,i}. \end{aligned} \tag{21}$$

We compute the rotations from the endpoint parameters of source trajectory i and anchor trajectory r. For readability, the checkpoint index T is omitted below. For the query-key pair, define

$$\begin{aligned} C_{qk,i}^h = W_{Q,i}^h \left(W_{Q,r}^h\right)^\top + W_{K,i}^h \left(W_{K,r}^h\right)^\top \\ + b_{Q,i}^h \left(b_{Q,r}^h\right)^\top + b_{K,i}^h \left(b_{K,r}^h\right)^\top. \end{aligned} \tag{22}$$

Let

$$C_{qk,i}^h = U_{qk,i}^h \Sigma_{qk,i}^h \left(V_{qk,i}^h\right)^\top \tag{23}$$

be its singular value decomposition. The optimal orthogonal transformation is

$$R_{qk,i}^h = U_{qk,i}^h \left(V_{qk,i}^h\right)^\top. \tag{24}$$

For the value-output pair, define

$$\begin{aligned} C_{vo,i}^h = W_{V,i}^h \left(W_{V,r}^h\right)^\top + b_{V,i}^h \left(b_{V,r}^h\right)^\top \\ + \left(W_{O,i}^h\right)^\top W_{O,r}^h. \end{aligned} \tag{25}$$

If

$$C^h_{vo,i} = U^h_{vo,i}\Sigma^h_{vo,i}\left(V^h_{vo,i}\right)^\top, \tag{26}$$

then

$$R^h_{vo,i} = U^h_{vo,i}\left(V^h_{vo,i}\right)^\top. \tag{27}$$

The two rotations are computed independently for every attention head. After alignment, the transformed head blocks are reassembled into the complete Q, K, V, and O projection matrices used in the spectral analysis. We use only the orthogonal rotation matching described above and do not apply an additional rescaling transformation.

A.2.3 Multi-Trajectory Alignment Strategies

Pairwise matching requires an anchor endpoint. For multiple trajectories, we consider the iterative strategy of Ainsworth et al. (2023) and the single-pass strategy used in our analysis. Table 2 summarizes their differences.

Table 2: Comparison of multi-trajectory alignment strategies. A pairwise alignment consists of the applicable permutation and attention-rotation matching operations.

Property	Iterative ensemble alignment	Subset-specific single-pass alignment
Reference	An ensemble reference updated from the aligned endpoints	The trajectory endpoint nearest to the subset endpoint mean
Update schedule	Repeatedly align endpoints and update the ensemble reference	Select the anchor once and align each remaining trajectory once
Number of passes	Multiple passes until the iterative matching procedure converges	One pass over the non-anchor trajectories
Pairwise alignments	Recomputed when the ensemble reference changes	$N-1$ pairwise alignments for a subset of size N
Coordinate system	Determined by the converged ensemble alignment	Determined by the selected trajectory $r(S)$

Under our single-pass strategy, for each selected subset $S = \{i_1, \ldots, i_N\}$, we first compute the element-wise mean of its endpoint parameters:

$$\overline{\theta}^{(p)}_S = \frac{1}{N}\sum_{i\in S}\theta^{(p)}_i(T), \qquad p \in \mathcal{P}, \tag{28}$$

where $\mathcal{P}$ indexes all parameter tensors in the model. The mean endpoint is used only for anchor selection.

For each trajectory, we compute

$$d_i(S) = \sum_{p\in\mathcal{P}}\left\|\mathrm{vec}\left(\theta^{(p)}_i(T) - \overline{\theta}^{(p)}_S\right)\right\|_2. \tag{29}$$

No normalization by tensor size or parameter norm is applied. The anchor is the trajectory with the smallest distance:

$$r(S) = \arg\min_{i\in S} d_i(S). \tag{30}$$

For every non-anchor trajectory $i \in S$, the pairwise permutation and rotation transformations are estimated from $\theta_i(T)$ and $\theta_{r(S)}(T)$. We denote their composition by $g^\star_{i,S}$. The anchor uses the identity transformation. Each endpoint-derived transformation is applied to every checkpoint:

$$\widetilde{\theta}_{i,S}(t) = g^\star_{i,S}\cdot\theta_i(t), \qquad t = 1, \ldots, T. \tag{31}$$

The mean endpoint, anchor, and transformations are recomputed for every subset S, making the aligned representation subset-dependent.

A.3 Mode-1 Trajectory-Wise Unfolding

Because alignment is recomputed for each trajectory subset S, the aligned and centered weights are formally subset-dependent. We write them as $W^{(\ell)}_{i,\mathrm{c},S}(t)$ in this appendix. The main text suppresses the subscript S for readability. We also omit the layer index ℓ below.

A.3.1 Output- and Input-Mode Constructions

The output-mode matrices follow Equations 3 and 4, with the subset index S restored on $X^{(\mathrm{out})}_{i,S}$. The feature index (b,t) may use any fixed bijective ordering of input coordinates and checkpoints; changing this ordering only permutes columns and preserves the singular values.

The input-mode construction retains the input coordinate as the row axis:

$$X^{(\mathrm{in})}_{i,S} \in \mathbb{R}^{d_{\mathrm{in}} \times (d_{\mathrm{out}} T)}, \qquad \left(X^{(\mathrm{in})}_{i,S}\right)_{b,(a,t)} = \left(W_{i,\mathrm{c},S}(t)\right)_{a,b}. \tag{32}$$

The corresponding multi-trajectory matrix is

$$M^{(1)}_{\mathrm{in}}(S) = \begin{bmatrix} X^{(\mathrm{in})}_{i_1,S} \\ X^{(\mathrm{in})}_{i_2,S} \\ \vdots \\ X^{(\mathrm{in})}_{i_N,S} \end{bmatrix} \in \mathbb{R}^{(N d_{\mathrm{in}}) \times (d_{\mathrm{out}} T)}. \tag{33}$$

The two constructions use the same centered trajectory tensor and differ only in which parameter coordinate is retained as the row axis.

A.3.2 Gram Matrix and Rank Bound

The output-mode Gram matrix expands as

$$\begin{aligned} & M^{(1)}_{\mathrm{out}}(S)^\top M^{(1)}_{\mathrm{out}}(S) \\ &= \begin{bmatrix} X^{(\mathrm{out})\top}_{i_1,S} & X^{(\mathrm{out})\top}_{i_2,S} & \cdots & X^{(\mathrm{out})\top}_{i_N,S} \end{bmatrix} \begin{bmatrix} X^{(\mathrm{out})}_{i_1,S} \\ X^{(\mathrm{out})}_{i_2,S} \\ \vdots \\ X^{(\mathrm{out})}_{i_N,S} \end{bmatrix} \\ &= \sum_{i \in S} X^{(\mathrm{out})\top}_{i,S} X^{(\mathrm{out})}_{i,S}. \end{aligned} \tag{34}$$

Similarly,

$$M^{(1)}_{\mathrm{in}}(S)^\top M^{(1)}_{\mathrm{in}}(S) = \sum_{i \in S} X^{(\mathrm{in})\top}_{i,S} X^{(\mathrm{in})}_{i,S}. \tag{35}$$

Temporal centering implies

$$\sum_{t=1}^{T} W_{i,\mathrm{c},S}(t) = 0 \qquad \text{for every } i \in S. \tag{36}$$

For each input coordinate in the output-mode construction, the T temporal columns therefore satisfy one linear constraint. All T checkpoints remain in the matrix, but they span at most $T-1$ independent temporal directions. Consequently,

$$\operatorname{rank}\left(M^{(1)}_{\mathrm{out}}(S)\right) \le \min\left(N d_{\mathrm{out}}, d_{\mathrm{in}}(T-1)\right). \tag{37}$$

The input-mode counterpart is

$$\operatorname{rank}\left(M^{(1)}_{\mathrm{in}}(S)\right) \le \min\left(N d_{\mathrm{in}}, d_{\mathrm{out}}(T-1)\right). \tag{38}$$

A.3.3 Invariance to Stacking Order

For a fixed subset S and fixed aligned matrices $\{X_{i,S}^{(\mathrm{out})}\}_{i\in S}$, changing only the trajectory stacking order corresponds to a row permutation. Let Q_π denote the associated permutation matrix. Then

$$M_{\pi,\mathrm{out}}^{(1)}(S) = Q_\pi M_{\mathrm{out}}^{(1)}(S). \tag{39}$$

Since $Q_\pi^\top Q_\pi = I$,

$$\left(M_{\pi,\mathrm{out}}^{(1)}(S)\right)^\top M_{\pi,\mathrm{out}}^{(1)}(S) = M_{\mathrm{out}}^{(1)}(S)^\top M_{\mathrm{out}}^{(1)}(S). \tag{40}$$

Thus, stacking order does not change the Mode-1 singular values in exact arithmetic. The same argument applies to the input-mode construction.

A.4 Mode-2 Global Unfolding

A.4.1 Output- and Input-Mode Constructions

The output-mode construction follows Equation 6, with the subset index S restored on each trajectory matrix.

The input-mode construction is

$$M_{\mathrm{in}}^{(2)}(S) = \begin{bmatrix} X_{i_1,S}^{(\mathrm{in})} & X_{i_2,S}^{(\mathrm{in})} & \cdots & X_{i_N,S}^{(\mathrm{in})} \end{bmatrix} \in \mathbb{R}^{d_{\mathrm{in}}\times(Nd_{\mathrm{out}}T)}, \tag{41}$$

with

$$\left(M_{\mathrm{in}}^{(2)}(S)\right)_{b,(n,a,t)} = \left(W_{i_n,\mathrm{c},S}(t)\right)_{a,b}. \tag{42}$$

Each input-mode row represents one aligned input coordinate across the selected trajectories, output coordinates, and checkpoints.

A.4.2 Gram Matrix and Rank Bound

The output-coordinate Gram matrix is

$$\begin{aligned} &M_{\mathrm{out}}^{(2)}(S)M_{\mathrm{out}}^{(2)}(S)^\top \\ &= \begin{bmatrix} X_{i_1,S}^{(\mathrm{out})} & X_{i_2,S}^{(\mathrm{out})} & \cdots & X_{i_N,S}^{(\mathrm{out})} \end{bmatrix} \begin{bmatrix} X_{i_1,S}^{(\mathrm{out})\top} \\ X_{i_2,S}^{(\mathrm{out})\top} \\ \vdots \\ X_{i_N,S}^{(\mathrm{out})\top} \end{bmatrix} \\ &= \sum_{i\in S} X_{i,S}^{(\mathrm{out})} X_{i,S}^{(\mathrm{out})\top}. \end{aligned} \tag{43}$$

For the input mode,

$$M_{\mathrm{in}}^{(2)}(S)M_{\mathrm{in}}^{(2)}(S)^\top = \sum_{i\in S} X_{i,S}^{(\mathrm{in})} X_{i,S}^{(\mathrm{in})\top}. \tag{44}$$

For each trajectory–input-coordinate pair in the output mode, temporal centering imposes one constraint across the T checkpoint columns. The column space therefore has dimension at most $Nd_{\mathrm{in}}(T-1)$. Since the matrix has d_{out} rows,

$$\mathrm{rank}\left(M_{\mathrm{out}}^{(2)}(S)\right) \le \min\left(d_{\mathrm{out}}, Nd_{\mathrm{in}}(T-1)\right). \tag{45}$$

The input-mode counterpart is

$$\mathrm{rank}\left(M_{\mathrm{in}}^{(2)}(S)\right) \le \min\left(d_{\mathrm{in}}, Nd_{\mathrm{out}}(T-1)\right). \tag{46}$$

The coordinate-space rank ceiling does not exceed d_{out} in the output mode or d_{in} in the input mode. This fixed ceiling does not imply that the effective dimensionality is constant. Adding trajectories changes the corresponding Gram matrix and its normalized spectrum.

A.4.3 Invariance to Trajectory Order

For a fixed subset and fixed aligned trajectory matrices, changing the trajectory order in Mode 2 corresponds to a column permutation. Let $\widehat{Q}_\pi$ denote the corresponding permutation matrix. Then

$$M^{(2)}_{\pi,\mathrm{out}}(S) = M^{(2)}_{\mathrm{out}}(S)\widehat{Q}_\pi. \tag{47}$$

Since $\widehat{Q}_\pi \widehat{Q}_\pi^\top = I$,

$$M^{(2)}_{\pi,\mathrm{out}}(S)\left(M^{(2)}_{\pi,\mathrm{out}}(S)\right)^\top = M^{(2)}_{\mathrm{out}}(S)M^{(2)}_{\mathrm{out}}(S)^\top. \tag{48}$$

The Mode-2 singular values are therefore invariant to trajectory order in exact arithmetic. The same result holds for the input mode.

This invariance should be distinguished from random prefix selection. For a random permutation π_b, the prefix

$$S_{b,N} = \{\pi_b(1), \ldots, \pi_b(N)\} \tag{49}$$

determines which trajectories enter the analysis. Different prefixes may produce different subsets and therefore different values of $D(S_{b,N})$.

When the full pipeline is rerun after reordering the trajectories, it also recomputes the endpoint mean, the subset-specific anchor, the pairwise alignment transformations, and the numerical SVD. Floating-point accumulation and numerical optimization can therefore produce small differences even when the selected trajectory set is unchanged. In our order-sensitivity checks, the resulting effective-dimensionality estimates differed by less than one. This numerical variation is distinct from the exact stacking-order invariance of the fixed matrices.

A.5 Summary of Representation Shapes

Table 3 summarizes the matrix and tensor representations used throughout the method. The table reports the output-mode shapes. For fully connected layers and attention projections, input-mode shapes follow by exchanging d_{out} and d_{in}. Convolutional layers instead use the input-channel matrixization defined in Appendix A.1.

Table 3: Shapes of the parameter and trajectory representations. Here $N = |S|$ and T is the number of checkpoints per trajectory.

Object	Shape	Construction
$W_i^{(\ell)}(t)$	$d_{\mathrm{out}}^{(\ell)} \times d_{\mathrm{in}}^{(\ell)}$	Matrixized weight of layer ℓ from trajectory i at checkpoint t.
$\mathcal{X}_S^{(\ell)}$	$d_{\mathrm{out}}^{(\ell)} \times d_{\mathrm{in}}^{(\ell)} \times T \times N$	Layer-wise tensor over parameter coordinates, checkpoints, and selected trajectories.
$X_{i,S}^{(\mathrm{out})}$	$d_{\mathrm{out}} \times (d_{\mathrm{in}}T)$	Output-mode unfolding of trajectory i after subset-specific alignment and temporal centering.
$M^{(1)}_{\mathrm{out}}(S)$	$(Nd_{\mathrm{out}}) \times (d_{\mathrm{in}}T)$	Mode-1 matrix formed by stacking the selected trajectory matrices vertically.
$M^{(2)}_{\mathrm{out}}(S)$	$d_{\mathrm{out}} \times (Nd_{\mathrm{in}}T)$	Mode-2 matrix formed by concatenating the selected trajectory matrices horizontally.

A.6 Trajectory Dataset Summary

Table 4 summarizes the trajectory collections used in our experiments. The public NiNo collections cover CNNs trained on image classification tasks and GPT2-style Transformers trained on language modeling. They provide densely sampled checkpoints, with 300 available trajectories for each CNN setting and 200 for each Transformer setting. The checkpoint interval specifies the spacing between saved model states, while T denotes the number of checkpoints in each trajectory.

Our DeBERTa-v3 collections consist of independently seeded fine-tuning runs from a shared pretrained initialization within each model size. The `xsmall` and `small` variants each cover eight

emotion datasets, whereas the `base` variant covers Grounded Emotions and Emotion-Cause. Each model-dataset pair contains 40 trajectories with 20 checkpoints saved once per epoch. The trajectory counts in the table describe the available collections; individual analyses use subsets of these trajectories according to the protocols specified in the main text and the corresponding ablations.

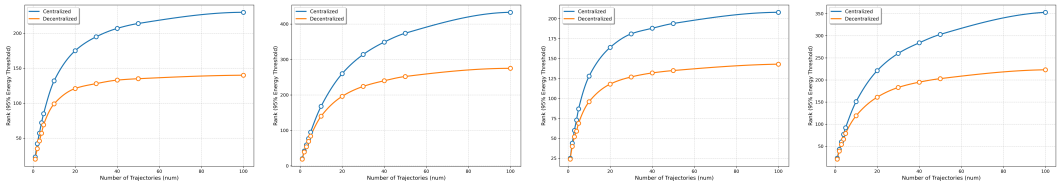
Table 4: Summary of training trajectory datasets. For DeBERTa-v3, the trajectory count is reported as the number of runs per dataset multiplied by the number of datasets.

Architecture	Dataset(s)	Trajectories	Checkpoint Interval	T
C10-16 (16-32-32)	CIFAR10	300	Every 4 Adam steps	2688
FM-16 (16-32-32)	FashionMNIST	300	Every 4 Adam steps	2513
Transformer (3-24-3)	LM1B	200	Every 200 Adam steps	124
Transformer (2-32-2)	LM1B	200	Every 200 Adam steps	124
DeBERTa-v3-`xsmall`	Emotion classification ($\times 8$)	40×8	Every epoch	20
DeBERTa-v3-`small`	Emotion classification ($\times 8$)	40×8	Every epoch	20
DeBERTa-v3-`base`	Grounded Emotions; Emotion-Cause	40×2	Every epoch	20

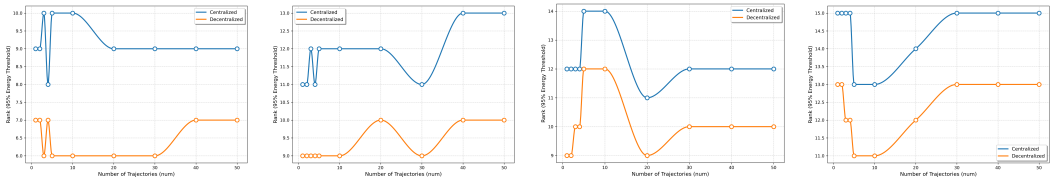
B Additional Ablations and Robustness Analyses

B.1 Effect of Temporal Centering

We examine the effect of temporal centering on the four NiNo trajectory collections. For each collection, we use a fixed trajectory ordering and compare centered and uncentered trajectories while holding the remaining preprocessing and spectral-analysis settings fixed. We plot $D_{0.95}(N)$ as the number of trajectories N increases. The CNN collections include up to 100 trajectories, while the Transformer collections include up to 50 trajectories. Figure 7 reports two representative layers from each collection.



(a) FM-16, Layer 1 (b) FM-16, Layer 2 (c) C10-16, Layer 1 (d) C10-16, Layer 2



(e) LM1B-3-24, Layer 1 (f) LM1B-3-24, Layer 2 (g) LM1B-2-32, Layer 1 (h) LM1B-2-32, Layer 2

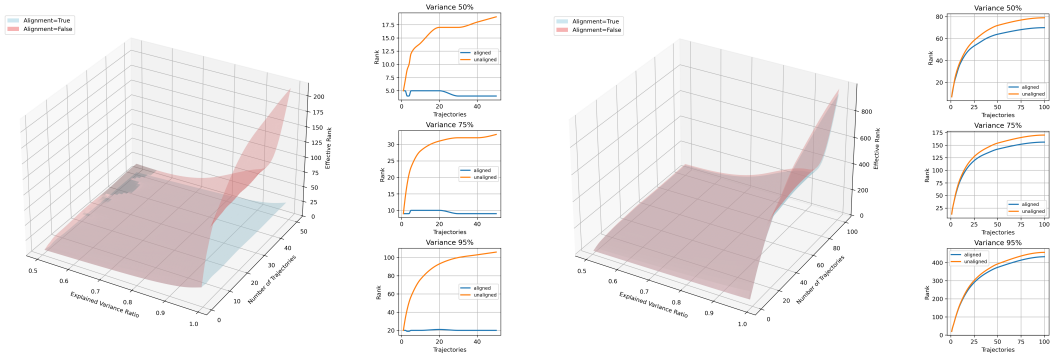
Figure 7: Effect of temporal centering on the four NiNo trajectory collections. Panels compare $D_{0.95}(N)$ with and without temporal centering at two representative layers per collection, using a fixed trajectory ordering. CNN and Transformer results extend to $N = 100$ and $N = 50$, respectively.

Temporal centering generally increases $D_{0.95}(N)$ in the displayed layers. Without centering, the trajectory-specific temporal mean remains in the matrix and can concentrate spectral energy in the leading directions. Subtracting this component distributes the energy over more directions. The dependence on N remains qualitatively similar under both settings over the observed range. The finite-range trend is therefore not solely determined by static parameter offsets.

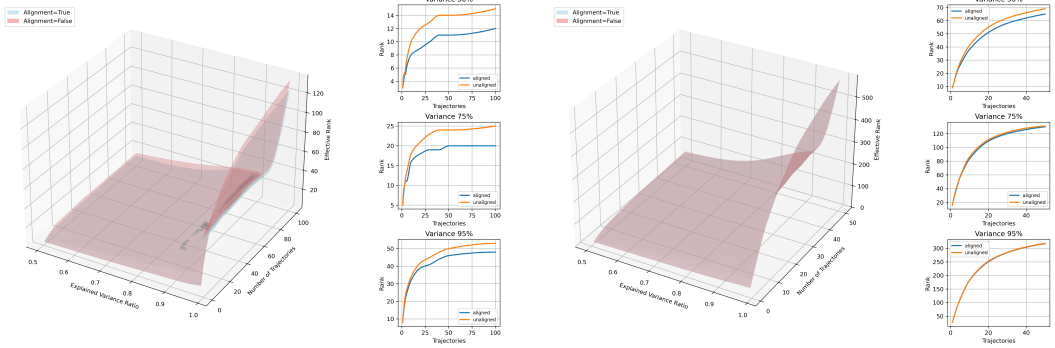
B.2 Effect of Symmetry Alignment

We examine how function-preserving coordinate symmetries affect the estimated effective dimension on the four NiNo trajectory collections. For each comparison, we use a fixed trajectory ordering and hold temporal centering, tensor unfolding, and spectral estimation fixed. Attention layers use rotation alignment. CNN, MLP, and Transformer FFN layers use permutation alignment.

Figure 8 reports four representative architectural components. Each panel shows $D_\tau(N)$ over the energy threshold τ and the number of trajectories N. The accompanying curves correspond to $\tau \in \{0.50, 0.75, 0.95\}$.



(a) Attention (b) CNN



(c) MLP (d) FFN

Figure 8: Effective dimension with and without symmetry alignment under a fixed trajectory ordering. Each panel contains a surface of $D_\tau(N)$ over τ and N, together with curves at $\tau \in \{0.50, 0.75, 0.95\}$. Alignment produces the largest reduction in effective dimension for attention layers. The changes are smaller for the CNN, MLP, and FFN examples.

Rotation alignment substantially reduces $D_\tau(N)$ for the attention example across the displayed thresholds. The unaligned trajectories include variation from function-preserving changes of basis. Alignment removes this coordinate redundancy and concentrates the spectral energy in fewer directions.

Permutation alignment produces smaller changes in the CNN, MLP, and FFN examples. Their aligned and unaligned curves retain a similar dependence on N. Both settings show flattening over the displayed range. The observed finite-range trend is therefore not introduced solely by the alignment procedure. We use aligned trajectories in the primary experiments to remove known rotation and permutation redundancies.

B.3 Layer-Wise Effective Dimension

We extend the default analysis in Section 4.1 to the first, middle, and last Transformer blocks of DeBERTa-v3-`xsmall` on the eight emotion datasets, using the same trajectory subsets at each layer position. Figure 9 compares both effective-dimension statistics across these positions.

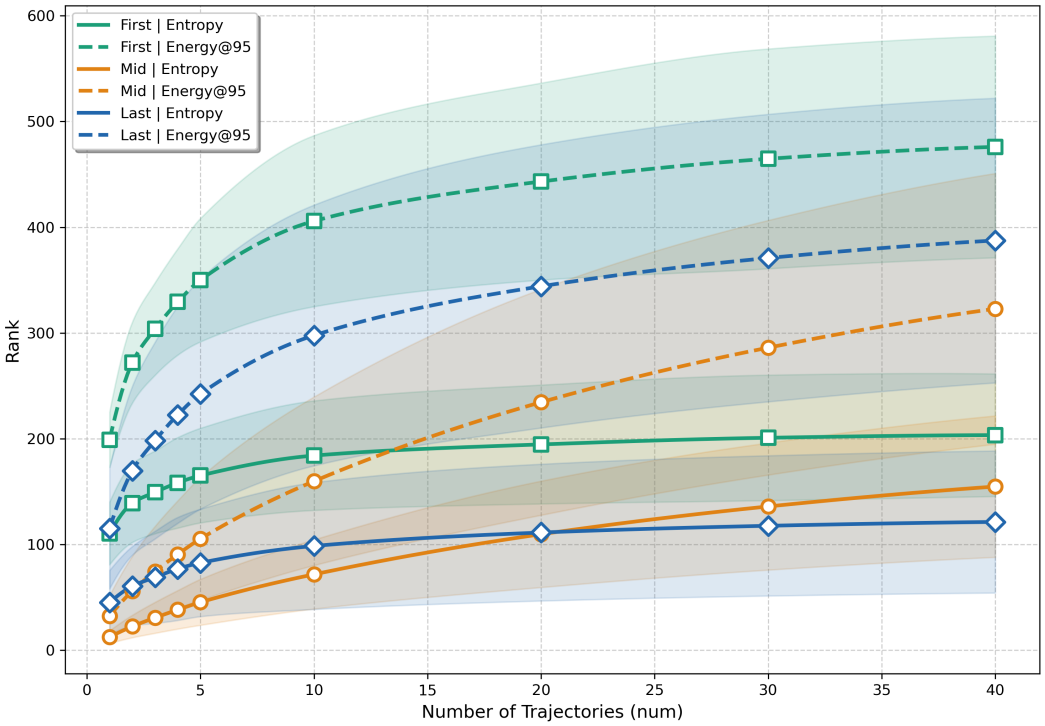


Figure 9: Layer-wise effective dimension on DeBERTa-v3-`xsmall`. Solid lines show the entropy-based estimate $\widehat{D}_{\mathrm{ent}}(N)$. Dashed lines show the energy-threshold estimate $\widehat{D}_{0.95}(N)$. Results are summarized separately for the first, middle, and last blocks, pooling trajectory orderings, the eight emotion datasets, and the eight projection-axis configurations. Lines and shaded bands show means and standard deviations, respectively.

The curves flatten over the observed range at all three layer positions. Their absolute values and relative ordering differ between $\widehat{D}_{\mathrm{ent}}(N)$ and $\widehat{D}_{0.95}(N)$. We therefore observe no consistent monotonic relationship between effective dimension and layer depth. The first-layer results used in the primary analysis capture a trend that is also present at the middle and last blocks, but the numerical scale varies across layers.

B.4 Robustness to Checkpoint Sampling Density

We examine whether the measured effective dimension depends on the temporal sampling density of the trajectories. This experiment uses DeBERTa-v3-`xsmall` fine-tuned on `Emotion-Cause`. From the same fine-tuning runs, we construct six trajectory variants by retaining checkpoints every 10, 20, 30, 40, or 50 optimization steps, or once per epoch. These variants contain 122, 61, 41, 31, 24, and 20 checkpoints per trajectory, respectively.

We use a fixed ordering of five trajectories and evaluate $N \in \{1, 2, 3, 4, 5\}$ without trajectory-order resampling. For each sampling density and N, we analyze the Q, K, V, and O projection matrices in the first, middle, and last Transformer blocks. Each parameter matrix is analyzed once with its output axis retained and once with its input axis retained. Table 5 reports the mean and standard deviation across these $3 \times 4 \times 2 = 24$ configurations.

Both statistics remain similar across the six sampling densities. At $N = 5$, reducing the number of checkpoints from 122 to 20 changes the mean $D_{0.95}(N)$ from 125.792 to 124.917 and the mean $D_{\mathrm{ent}}(N)$ from 43.223 to 42.455. These changes are 0.70% and 1.78%, respectively. The reported standard deviations describe variation across layers, projections, and retained axes rather than trajectory-order uncertainty. The effective-dimension estimates are therefore stable across the tested checkpoint sampling densities.

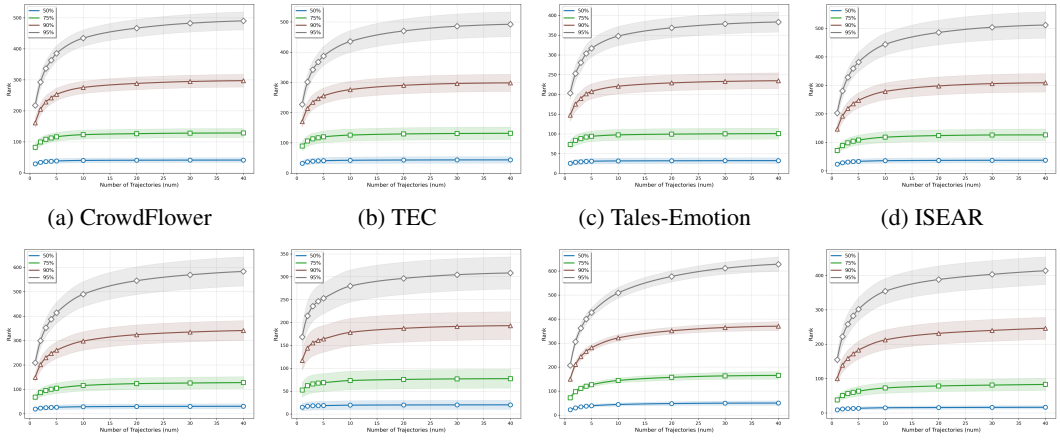
Table 5: Effective dimension under different checkpoint sampling densities on DeBERTa-v3-xsmall fine-tuned on Emotion-Cause. Entries report means ± standard deviations across 24 configurations: the eight projection-axis configurations at each of three layer positions. The upper block reports $D_{0.95}(N)$, and the lower block reports $D_{\text{ent}}(N)$. A fixed trajectory ordering is used without order resampling.

	Number of trajectories N				
Sampling density	1	2	3	4	5
Energy-threshold dimension $D_{0.95}(N)$					
Every 10 steps (122)	63.0 ± 63.5	88.6 ± 85.3	105.2 ± 96.9	116.8 ± 105.1	125.8 ± 108.6
Every 20 steps (61)	63.0 ± 63.5	88.6 ± 85.1	105.1 ± 96.8	117.0 ± 105.0	125.8 ± 108.3
Every 30 steps (41)	62.9 ± 63.3	88.4 ± 84.8	105.0 ± 96.5	116.8 ± 104.6	125.7 ± 108.1
Every 40 steps (31)	62.9 ± 63.3	88.4 ± 84.7	104.8 ± 96.2	116.7 ± 104.5	125.5 ± 107.6
Every 50 steps (24)	62.8 ± 63.0	88.2 ± 84.5	104.8 ± 95.8	116.5 ± 104.2	125.4 ± 107.5
Every epoch (20)	62.6 ± 62.8	87.9 ± 83.9	104.3 ± 95.3	116.2 ± 103.6	124.9 ± 106.8
Entropy-based dimension $D_{\text{ent}}(N)$					
Every 10 steps (122)	23.6 ± 23.3	31.6 ± 29.4	36.4 ± 30.9	40.3 ± 33.4	43.2 ± 33.3
Every 20 steps (61)	23.5 ± 23.2	31.5 ± 29.2	36.2 ± 30.7	40.2 ± 33.2	43.1 ± 33.1
Every 30 steps (41)	23.4 ± 23.0	31.4 ± 29.0	36.1 ± 30.5	40.0 ± 33.0	43.0 ± 32.8
Every 40 steps (31)	23.4 ± 22.9	31.4 ± 28.9	36.0 ± 30.3	40.0 ± 32.9	42.9 ± 32.7
Every 50 steps (24)	23.3 ± 22.8	31.2 ± 28.7	35.9 ± 30.1	39.8 ± 32.7	42.7 ± 32.5
Every epoch (20)	23.1 ± 22.5	31.0 ± 28.4	35.6 ± 29.7	39.6 ± 32.3	42.5 ± 32.1

B.5 Sensitivity to Spectral Statistics

We assess sensitivity to the spectral statistic using the same DeBERTa-v3-xsmall Mode-1 results as the main analysis on the eight emotion datasets. Both figures below follow the default per-dataset aggregation protocol in Section 4.1, reporting means and standard deviations.

Energy-Threshold Effective Dimension. Across the eight datasets, the mean curves $\widehat{D}_\tau(N)$ for $\tau \in \{0.50, 0.75, 0.90, 0.95\}$ differ in magnitude but show diminishing growth after the first few trajectories (Figure 10). The observed trend is therefore not specific to the 0.95 cutoff.



(a) CrowdFlower (b) TEC (c) Tales-Emotion (d) ISEAR

(e) EmoInt (f) ElectoralTweets (g) Grounded Emotions (h) Emotion-Cause

Figure 10: Energy-threshold effective dimension for $\tau \in \{0.50, 0.75, 0.90, 0.95\}$ across the eight emotion datasets. Curves show $\widehat{D}_\tau(N)$, and shaded bands show the corresponding standard deviations.

The separation between thresholds reveals how spectral energy is distributed across directions. For CrowdFlower at $N = 40$, the mean dimensions are 41.3, 128.6, 297.4, and 490.2 at thresholds

0.50, 0.75, 0.90, and 0.95, respectively. The much larger dimension required at higher thresholds is consistent with a concentrated leading spectrum and an extended lower-energy tail.

The late-range changes also depend on the threshold. Between $N = 30$ and $N = 40$, the average change per added trajectory is below 0.05 at $\tau = 0.50$ on every dataset, compared with 0.38–1.66 at $\tau = 0.95$. The increase remains larger at the higher cutoff, although both are small relative to their early-range growth.

Entropy-Based Effective Dimension. Figure 11 shows the complete dataset-wise curves for $\widehat{D}_{\text{ent}}(N)$, which exhibit the same finite-range slowing as the threshold measures.

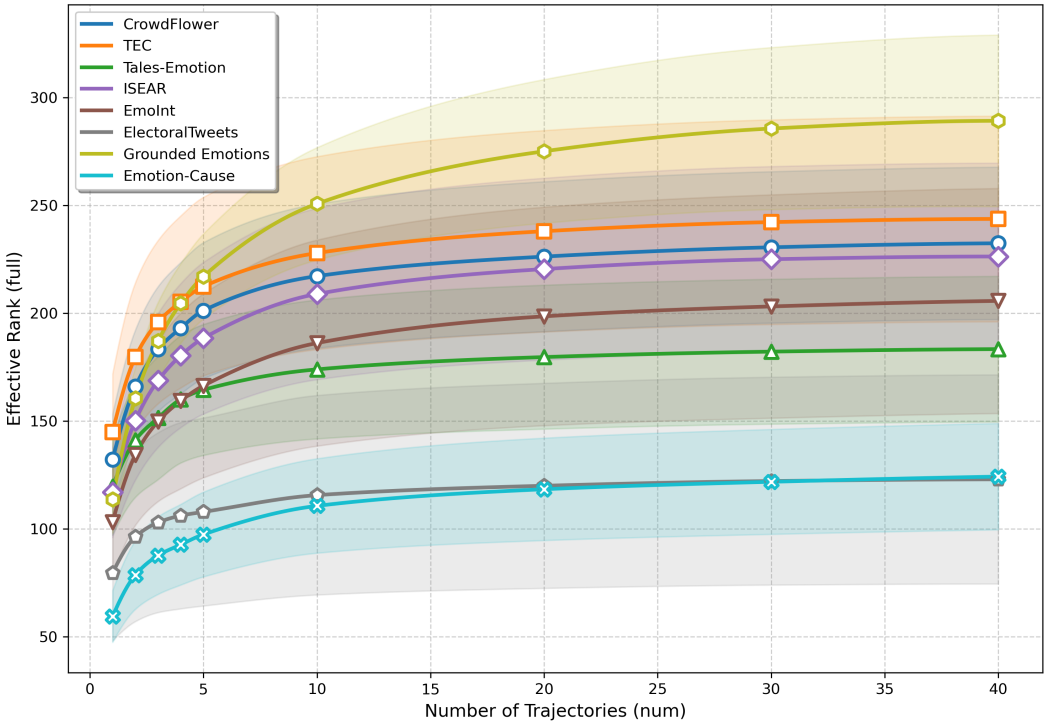


Figure 11: Entropy-based effective dimension across the eight emotion datasets. Curves show $\widehat{D}_{\text{ent}}(N)$, and shaded bands show the corresponding standard deviations.

The two statistics can rank datasets differently. At $N = 40$, EmoInt has a higher mean $D_{0.95}$ than CrowdFlower (583.1 versus 490.2), but a lower mean D_{ent} (205.7 versus 232.4). Thus, the statistics agree on diminishing growth while capturing different aspects of spectral concentration.