IT'S ALL IN THE HEADS: AN INVESTIGATION OF DOMAIN KNOWLEDGE INFUSION INTO LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

While large language models (LLMs) are widely studied, the mechanisms by which they internalize knowledge from specialized domains remain poorly understood. To investigate this, we analyze the Continual Pre-Training (CPT) paradigm, where a base model is further pre-trained on a curated, domain-specific corpus. Through a focused study on mathematical data, we uncover two key properties of this process: (1) domain connectivity between checkpoints trained on different CPT datasets, and (2) head-wise sparsity in the model increment that encodes new domain knowledge. We further support these findings with a spectral analysis of weight matrices at different lengths of pre-training stage before and after CPT, and investigate applicability of the heavy-tailed self-regularization theory to modern large language models. To foster further research, we provide an open-source scalable toolkit for performing spectral analysis on models with billions of parameters - NetInspect. The code is available at https://anonymous.4open.science/r/netinspect-EF67

1 INTRODUCTION

Continual pre-training (CPT) is now a standard component of modern LLM training pipelines; in many contemporary multi-stage workflows — including recent state-of-the-art models such as Llama 3 (Dubey et al., 2024) and OLMo 2 (OLMo et al., 2024) — the final pre-training stage uses curated, domain-specific data mixtures while the learning rate is linearly annealed to zero. Empirical evidence suggests that this late-stage focus on cleaner, domain-relevant data improves mathematical and coding abilities without degrading general-language performance (Blakeney et al., 2024). Moreover, CPT is central to producing domain-specialized models, such as Code Llama (Roziere et al., 2023) and DeepSeekMath (Shao et al., 2024); applying CPT to a general-purpose model yields better performance than training a specialized model from scratch under the same compute budget.

However, the CPT stage of the LLM training pipeline is significantly less studied compared to the supervised fine-tuning (SFT) stage, where a rich set of phenomena has been documented — including linear mode connectivity (Frankle et al., 2020), task arithmetic (Ilharco et al., 2023), model soups (Wortsman et al., 2022), ability transfer (Yu et al., 2024) and low-rank subspace modification (Hu et al., 2022). To investigate whether similar phenomena exist for CPT, we conduct a series of pre-train and continual pre-training experiments on 1B-parameter language models with OLMo 2 architecture and analyze the weight delta $\Delta \boldsymbol{W} = \boldsymbol{W}^{\text{math}} - \boldsymbol{W}^{\text{pre-train}}$, characterizing its sparsity and subspace geometry, and assessing its ability to interpolate between, and to merge, checkpoints adapted to different domains.

Additionally, we investigate singular spectra of weight matrices: prior random matrix theory-based work suggests that heavy-tailed singular value distributions are closely connected to generalization ability (Martin & Mahoney, 2019; 2021; Yang et al., 2022). In our spectral analysis, we do not limit ourselves to a single scalar metric, and we investigate the role of singular vectors, which were earlier shown to play an important role in the model training process (Yunis et al., 2024).

In summary, our contributions include:

1. We investigate the dynamics of weight matrix singular values spectra, and identify the development of complex spectral structure in attention head matrices along the pre-train stage, which can not be described by heavy-tailed self-regularization theory (Martin &

Mahoney, 2019). This is associated with an increase in quality on language tasks and faster domain adaptation on CPT. For CPT we find that the spectra remain almost stable, and the domain adaptation is driven by singular vector changes localized near the peaks in the SVD spectrum.

2. We identify the ability to interpolate between checkpoints after CPT on different domains, for which we coin the term *domain connectivity*; we find that interpolated model quality improves with increase in pre-train stage length. Additionally, we discover that CPT deltas have redundancy in parameters - we can drop up to 35% heads or 20% lowest singular values without significant changes to model quality.
3. We provide an open-source implementation of our matrix spectra analysis tools – NetInspect package – to ensure the reproducibility of our findings and to facilitate future research.

2 METHODOLOGY

In order to investigate the properties of model weight matrices, we employ several analytical methods centered on singular value decomposition (SVD) (Golub & Reinsch, 1970). For a weight matrix from the i-th layer $\boldsymbol{W}^i$, with dimensions $m \times n$ ($m \geq n$) and hard rank $r = R(\boldsymbol{W})$, its thin SVD is $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, where $\boldsymbol{U} = [\,\boldsymbol{u}_1(\boldsymbol{W}), \ldots, \boldsymbol{u}_r(\boldsymbol{W})\,] \in \mathbb{R}^{m\times r}$, $\boldsymbol{V} = [\,\boldsymbol{v}_1(\boldsymbol{W}), \ldots, \boldsymbol{v}_r(\boldsymbol{W})\,] \in \mathbb{R}^{n\times r}$, and $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1(\boldsymbol{W}), \ldots, \sigma_r(\boldsymbol{W}))$ with $\sigma_1(\boldsymbol{W}) \geq \cdots \geq \sigma_r(\boldsymbol{W}) > 0$. This notation is used consistently throughout our analysis.

Norms and Ranks. We characterize singular spectra $\boldsymbol{\Sigma}(\boldsymbol{W})$ through established spectral measures: the Frobenius norm $\|\mathbf{W}\|_F$, spectral norm $\|\boldsymbol{W}\|_2$, stable rank $R^s(\boldsymbol{W})$, and effective rank $R^e(\boldsymbol{W})$. Complete definitions are provided in Appendices F.1.1 and F.1.2.

Singular Vector Agreement. SVD provides both spectral magnitudes and directional information through singular vectors. To analyze directional changes during training, we measure agreement between singular vectors of a given weight matrix along the training trajectory. Further implementation details are presented in Appendix F.1.3.

Fitting Model Distributions. We model the empirical spectral density (ESD) using two complementary approaches: the Marchenko–Pastur distribution for the bulk spectrum and power-law models for heavy-tailed spectral regions. Estimation procedures and diagnostic methods are detailed in Appendix F.1.4.

3 EXPERIMENTAL SETUP

3.1 TRAINING SETUP

We initialize all models using the OLMo 2 architecture and training stack (OLMo et al., 2024), training 1B parameter model. Our experimental pipeline consists of two sequential stages:

1) **Stage 1: Pre-training.** Models are pre-trained from scratch on mixtures drawn from DCLM (Li et al., 2024), with token budgets varying from 20B to 400B. Training uses a cosine learning rate scheduler with a warm-up phase.

2) **Stage 2: Continual pre-training (CPT).** Starting from pre-trained checkpoints, we continue training on alternative data mixtures to probe domain shift and replay effects. We systematically vary the CPT data composition to emphasize: (i) DolminoMath-only data (OLMo et al., 2024), (ii) balanced DCLM+DolminoMath mixtures, or (iii) DCLM-heavy replay. The learning rate is initialized from the final value of Stage 1 and annealed to zero throughout this stage.

Complete reproducibility details, including hyperparameters, training configurations, and data splits, are provided in Appendix A.2. The code can be found in Appendix B.

3.2 EVALUATION PROTOCOL

The quality of the model is evaluated using the OLMES framework (Gu et al., 2025). Language accuracy is assessed by averaging results from three datasets: ARC-Easy (Clark et al., 2018), Hel-

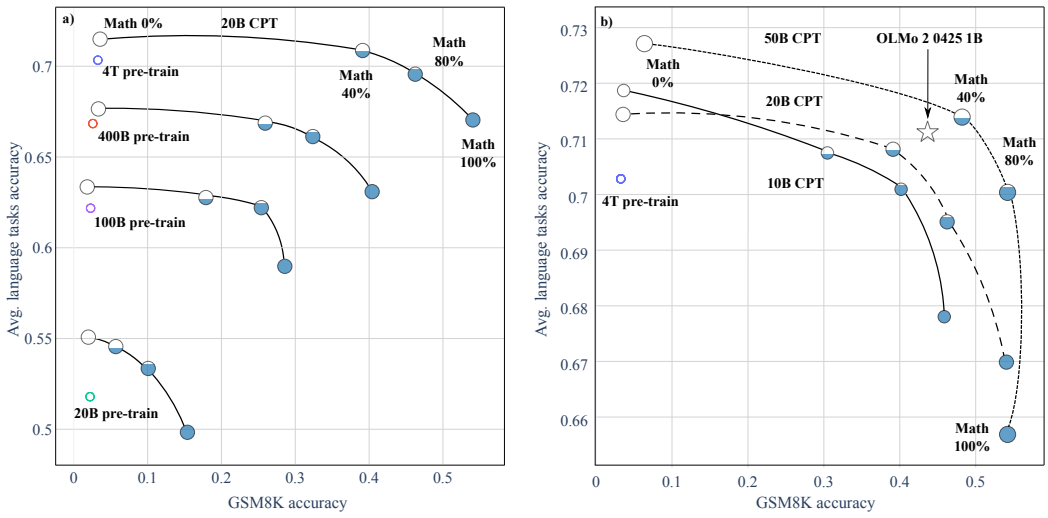

Figure 1: Math–language quality trade-off with continual pre-training on the OLMo 2 1B backbone. Y-axis reports the average accuracy on WinoGrande, ARC-Easy, and HellaSwag. Hollow circles denote pre-train checkpoints; filled circles denote continual pre-training runs initialized from the corresponding pre-train checkpoints. Marker size encodes the CPT token budget (10B, 20B, 50B); marker fill encodes the math proportion in CPT data (0%, 40%, 80%, 100%). Points with the same CPT token budget are connected. a) Overview across pre-train sizes (20B, 100B, 400B, 4T). For each pre-train checkpoint, we plot CPT runs with a fixed 20B-token budget (equal marker sizes) and varying math proportions; the leftmost hollow marker in each connected series is the corresponding pre-train checkpoint. b) Quality of CPT runs starting from 4T pre-train checkpoint (leftmost hollow marker). CPT runs vary both the token budget and the math proportion; the star marks the original OLMo 2 0425 1B CPT model (50B tokens with 10B math, i.e., 20%). Larger pre-train token budgets yield better results overall. Increasing the math proportion moves models rightward while typically lowering Language, whereas larger token budgets shift the trade-off frontier outward.

laSwag (Zellers et al., 2019), and WinoGrande (Sakaguchi et al., 2019). Each language dataset measures 5-shot accuracy by scoring each answer choice individually based on LLM token probabilities in a cloze format. For math accuracy, we use an 8-shot evaluation on GSM8k (Cobbe et al., 2021), calculating exact matches between the LLM-generated answers and the gold standard.

4 Main Results

4.1 Quality metrics

When comparing quality after CPT for models pre-trained with different compute budgets, we find that larger budgets enhance model flexibility (Figure 1 a). While initial math performance is similar across budgets, models with higher pre-training budgets achieve better final math scores after continual pre-training. These models also show faster increase in math performance as a function of CPT token budget, but experience more rapid declines in language accuracy. For instance, maintaining 80% math and 20% text token ratio with a 100B token budget preserves language performance, whereas the same ratio at a 4T budget results in greater language decline. This indicates that larger models adapt quickly in specialized tasks like math at the expense of general language skills, and reveals a trade-off where higher pre-training budgets boost target domain performance while risking broader language ability.

Panel (b) in Figure 1 illustrates that maintaining the same math-to-text token ratio while increasing total tokens enhances math abilities. However, for fully math-focused samples at 20B and 50B, math performance plateaus, yielding similar results. Notably, more math tokens lead to greater declines in language performance, highlighting the risks of over-specialization.

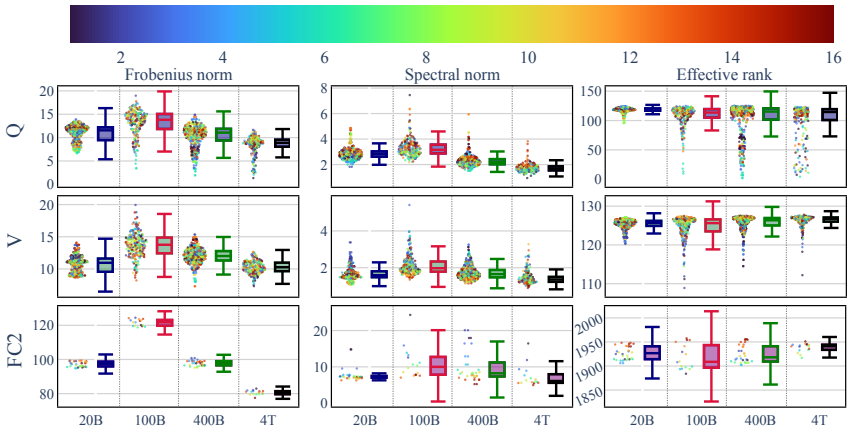

Figure 2: Pre-train evolution via spectral metrics. Rows (top to bottom) correspond to $\boldsymbol{W}^{\mathrm{Q}}$, $\boldsymbol{W}^{\mathrm{V}}$, and $\boldsymbol{W}^{\mathrm{FC_2}}$ weight matrices, while columns (left to right) report the Frobenius norm, the spectral norm, and the effective rank. Each subplot overlays jittered per-matrix values with box plots summarizing the distributions for models pre-trained on 20B, 100B, 400B, and 4T tokens. Points denote individual matrices and are color-coded by the corresponding layer index. Note non-monotonic behavior of matrix norms and different trajectories of effective rank along pre-train.

4.2 Spectral evolution along the pre-train stage

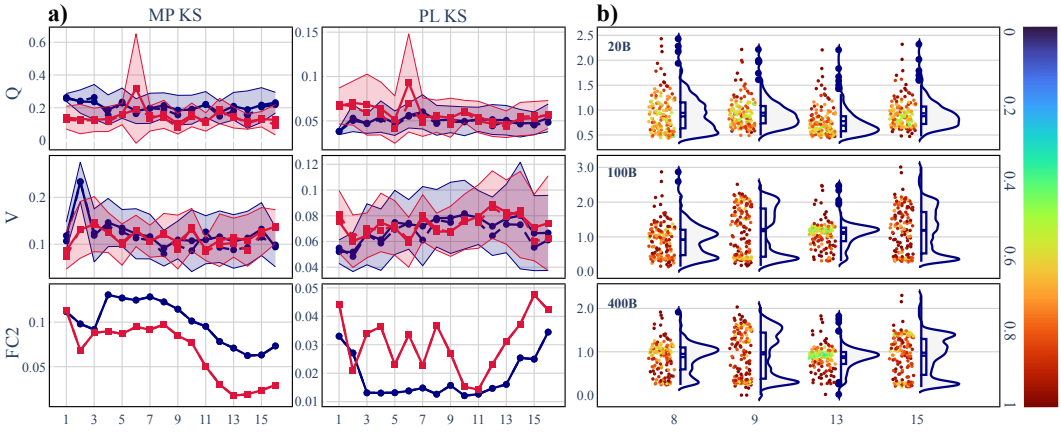

Figure 3: Spectral shape formation during pre-training. a) Goodness-of-fit - layer-wise Kolmogorov–Smirnov distance between the weight matrix ESD and Marchenko–Pastur model (left) and power-law (right) model for $\boldsymbol{W}^{\mathrm{Q}}$, $\boldsymbol{W}^{\mathrm{V}}$, and $\boldsymbol{W}^{\mathrm{FC_2}}$ matrices. The x-axis indexes layers; solid lines show the mean across attention heads, dashed lines the median, and the shaded band denotes mean ± std; blue denotes 20B-token pre-train and red denotes 400B-token pre-train. b) Singular-value spectra across pre-train for $\boldsymbol{W}^{\mathrm{Q}}$ layer 6, heads 8, 9, 13, 15; for 20B, 100B, and 400B pre-train tokens. Jitter points represent singular values and are colored by the left singular vector agreement between each pre-train checkpoint and the corresponding CPT endpoint on math domain. Note the increasingly complex spectral shape that poorly conforms to MP and PL models. Starting at 100B the singular vectors, that change direction during the CPT stage, are increasingly localized in narrow spectral bands.

To gain insight into pre-train dynamics, we first consider weight matrix norms and ranks (Figure 2). We highlight non-monotonic behavior of Frobenius and spectral norms, which reach maximum values at around 100B tokens of pre-train. Effective rank also demonstrates an inflection point around 100B tokens: a substantial number of layer-level outliers with significantly lower rank emerges. However, its behavior differs for different matrix types: for $\boldsymbol{W}^{\mathrm{Q}}$, $\boldsymbol{W}^{\mathrm{K}}$ the outliers are severe, the

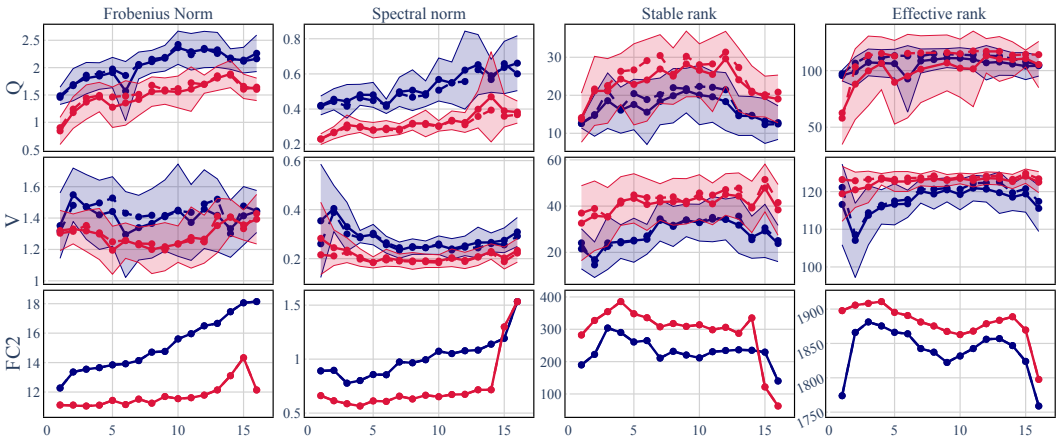

Figure 4: Layer-wise spectral analysis (Frobenius norm, spectral norm, stable rank and effective rank) of CPT deltas for $\boldsymbol{W}^{\mathrm{Q}}$, $\boldsymbol{W}^{\mathrm{V}}$, and $\boldsymbol{W}^{\mathrm{FC_2}}$. Deltas from models pre-trained on 100B tokens (blue) exhibit higher Frobenius and spectral norms but lower stable and effective ranks compared to those from 4T tokens (red). Statistics are aggregated per head (mean: solid line, median: dashed line; shaded region: mean ± std).

amount of outliers increases, mean rank decreases, while for $\boldsymbol{W}^{\mathrm{V}}$, $\boldsymbol{W}^{\mathrm{O}}$ and MLP matrices outliers tend to recover rank after 100B tokens.

Next, we investigate the applicability of power law and Marchenko-Pastur model spectra, in accordance with heavy-tail self-regularization theory (Figure 3). We note that for weight matrices after 20B and 400B tokens pre-train budget both model spectra exhibit poor fit quality for attention-related matrices, while MLP-related matrices show decent accordance with power-law tail fit. To investigate this further, we examine SVD spectra for several heads of $\boldsymbol{W}^{\mathrm{Q}}$ along the pre-train stage. We note the increasing complexity of spectral shape from 20B to 400B tokens. We conjecture that complex spectral structure is a prerequisite for fine-grained adjustments to matrix structure, and may explain higher math accuracy after CPT for models with longer pre-train stages. We leave detailed investigation for further work.

4.3 Spectral evolution along the CPT stage

To understand the mechanisms of continual pre-training, we analyze how the spectral features of the model evolve from pre-train to CPT.

Our analysis reveals that the CPT stage does not change matrix spectra appreciably, and the changes associated with the increase in GSM8K accuracy are due to the changes in singular vectors (see Appendix E). Analysis of vector agreement shows that it is the highest in early layers and decreases toward the output layers. The same dynamics applies to Frobenius norm of the CPT deltas $\Delta\boldsymbol{W} = \boldsymbol{W}^{\text{math}} - \boldsymbol{W}^{\text{pre-train}}$. Additionally, we find that CPT deltas are high-rank across all matrix types.

While these structural patterns are consistent, the *norm and rank* of deltas vary with pre-training budget. As illustrated in Figure 4, CPT starting from smaller pre-train budget of 100B tokens produces delta with substantially larger Frobenius and spectral norms than from 4T tokens, indicating necessity of more extensive parameter adjustments. However, the stable and effective ranks of these deltas are lower (can be twice lower for stable rank) indicating that updates from earlier checkpoints, while larger in magnitude, are confined to narrower subspaces.

In order to validate singular vector agreement as a metric for assessing the change during CPT, we consider CPT trajectories on 20B tokens datasets with varying composition: math, text, and a mixture of 8B math and 12B text. Agreement for these trajectories is shown in Figure 5. We observe a clear monotonic relationship: as the CPT domain shifts from text (aligned) to math (divergent), overlap with pre-train decreases correspondingly. Vector agreement can serve as a more detailed

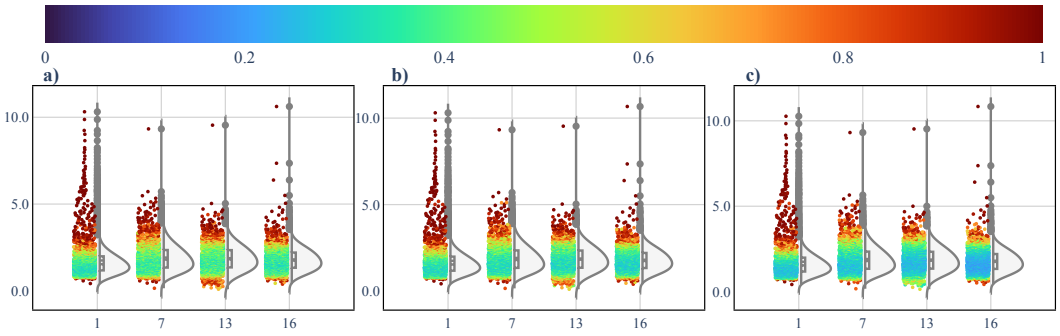

Figure 5: The effect of CPT data domain on vector agreement with pre-train. Violin plots (with jittered points colored by the left singular vector agreement between CPT and pre-train) display the $\boldsymbol{W}^{\text{GATE}}$ per layer for CPT a) on text, b) a math-text mix, c) or pure math (all pre-trained on 4T tokens). The results demonstrate a clear ordering: text CPT preserves the strongest vector agreement, while math domain induces the largest rotate.

description of the induced changes compared to delta norms, since it highlights the position of changing vectors in the singular spectrum.

4.4 Domain connectivity

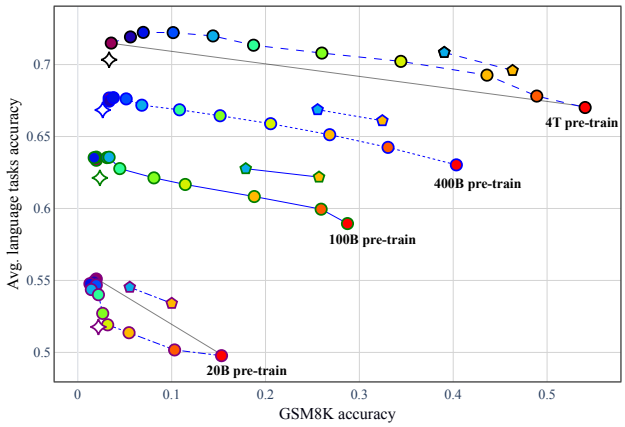

Figure 6: Interpolation quality: average language task accuracy vs. GSM8K accuracy for different pre-train token budgets against a filtered DCLM+DolminoMath mixture, step size $\omega = 0.1$. Yellow markers correspond to 80% Math / 20% filtered DCLM; blue markers correspond to 40% Math / 60% filtered DCLM. As the pre-train token budget increases, the distance between the yellow circle and yellow pentagon decreases, indicating that interpolation quality approaches that of joint training on the mixture.

Inspired by linear mode connectivity for fine-tuning of large language models and vision models, we observe a similar phenomenon for continual pre-training. We study domain connectivity between two models initialized from the same $\boldsymbol{W}^{\text{pre-train}}$ checkpoint and trained on 20B CPT datasets: DolminoMath $\boldsymbol{W}^{\text{math}}$ and filtered DCLM $\boldsymbol{W}^{\text{text}}$. We form interpolants

$$\boldsymbol{W}^{\text{interp}}(\omega) \;=\; (1-\omega)\,\boldsymbol{W}^{\text{text}} \;+\; \omega\,\boldsymbol{W}^{\text{math}}, \qquad \omega \in [0,1], \tag{1}$$

which we call "*model soups*", and compare their quality to models trained on 8BDM and 16BDM data mixes (Table 2).

Across pre-train budgets, linear interpolation underperforms CPT trained on dataset mixtures, but the performance gap shrinks consistently as the pre-training token budget increases (see 6). In the

(GSM8K accuracy, mean accuracy across language tasks) space, the model soup quality curve lies *below the chord* connecting the endpoints (i.e., is concave) for 20B, is approximately *linear* at 100B, becomes slightly *convex* at 400B, and is clearly *convex* at 4T. This phenomenon might relate to the decrease in Frobenius norm between math and text checkpoints with the pre-train budget (Appendix Table 4) and more complex spectral structure of the attention heads of later checkpoints. We leave detailed investigation of this effect to further work.

4.5 Localizing specialization and redundancy

Next, we investigate where domain specialization in $\boldsymbol{W}^{\text{math}}$ resides and how redundant the updates are. We study two complementary manipulations: (i) selectively *rewinding* attention heads to their pre-train values, and (ii) truncating the singular spectrum of the CPT delta.

Head-level rewind. For 20B CPT on DolminoMath, we rewind parameters of a single attention head (including the corresponding QK-norm components) back to the pre-train checkpoint while leaving the rest of the network unchanged. We apply this procedure to both 100B- and 4T-long pre-trains of the 1B-parameter model. Rewinding any single head changes GSM8K accuracy and average language tasks accuracy by less than one percentage point on average, and for some heads can *improve* GSM8K accuracy (see Fig. 12 in Appendix E). This supports the hypothesis that domain adaptation is distributed across multiple heads rather than concentrated in a few.

We order each head by its single-head rewind impact and then rewind heads in descending (greedy) order. The resulting curves dominate those produced by static heuristics based on spectral/RMT-style proxies—delta Frobenius norm, delta stable rank, pre-train PL–KS, or $\boldsymbol{W}^{\text{pre-train}} \leftrightarrow \boldsymbol{W}^{\text{math}}$ singular-vector overlap—which perform near a random baseline. Thus, simple spectral scalars are insufficient to predict a head's causal importance for CPT quality gains (see Fig. 7 a). This is in accordance with the complex shape of attention matrices spectra, and requires further research into devising appropriate model distributions.

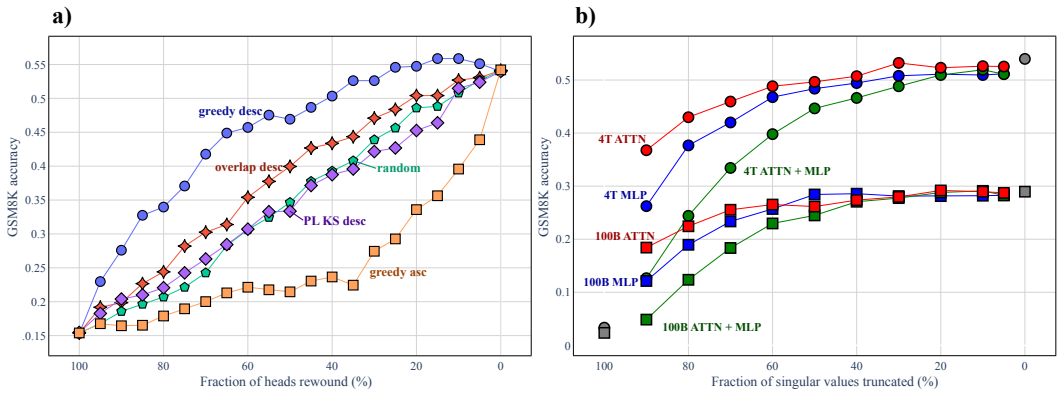

Figure 7: a) GSM8K accuracy versus fraction of heads rewound. Greedy ordering strongly outperforms heuristic proxies. b) GSM8K accuracy versus fraction of singular values kept under delta SVD truncation.

SVD truncation of CPT delta. To assess redundancy, we analyze the low-rank structure of the CPT delta $\Delta\boldsymbol{W} = \boldsymbol{W}^{\text{math}} - \boldsymbol{W}^{\text{pre-train}}$. For each matrix, we compute an SVD of $\Delta\boldsymbol{W}$, zero out the smallest singular values, reconstruct a truncated $\Delta\widetilde{\boldsymbol{W}}$, and evaluate $\boldsymbol{W}^{\text{pre-train}} + \Delta\widetilde{\boldsymbol{W}}$. Truncation is applied only to Attention (head-wise) and MLP matrices; all other matrices remain from $\boldsymbol{W}^{\text{math}}$.

Deltas are high-rank, but allow removal of up to $\mathbf{20}\%$ of singular values yields no measurable GSM8K accuracy drop, while pronounced degradation begins at $\geq \mathbf{70}\%$ removal. The same thresholds hold qualitatively at 100B initialization.

We further study the effect of truncation in attention and MLP matrices. At 60% truncation within Attention, GSM8K accuracy drops by only $\approx \mathbf{10}\%$; MLP truncation decreases the quality more, and truncating both attention and MLP hurts the most. In the extreme regime with only 10% singular values kept in the difference, we observe the following decreases in quality: for attention matrices:

$-\mathbf{30}\%$, MLP: $-\mathbf{52}\%$, and attention+MLP: $-\mathbf{74}\%$. Thus, the $\boldsymbol{W}^{\text{math}}$ signal is more truncation-tolerant in attention than in MLPs. The qualitative ordering also holds at 100B initialization.

With the above experiments, we find that domain knowledge is distributed across many singular directions. Nevertheless, a meaningful fraction of the delta lies in a compressible tail, enabling moderate rank reduction without quality loss — especially in attention matrices (see Fig. 7; Appendix Table 7).

5 Related work

Continual pre-training. A primary challenge in CPT is mitigating catastrophic forgetting while efficiently adapting models to new domains. To combat forgetting directly, interleaving a small fraction of *replay data* from the original pre-training distribution during CPT is a simple yet powerful method to anchor the model's general representations (Wang et al., 2023; Qi et al., 2025; Hickok, 2025). The optimization process itself is crucial, as empirical findings demonstrate that learning rate re-warming and re-decaying is necessary to overcome initial instability and adapt optimization dynamics to the new data, even in the absence of a distribution shift (Gupta et al., 2023; Ibrahim et al., 2024). Furthermore, data selection strategies that choose samples based on their similarity to the target task or their novelty and diversity are highly effective for adaptation (Xie et al., 2023; Que et al., 2024).

Model weight spectrum interventions. Weight matrices are often approximately low rank, enabling selective removal of higher-order components (small singular values) to denoise networks and enhance reasoning performance (Sharma et al., 2023). This low-rank property underpins parameter-efficient fine-tuning methods like LoRA (Hu et al., 2022). However, optimal rank is highly layer-dependent, and smaller residual singular values are not mere noise—they maintain connectivity to good loss basins and preserve performance on difficult tasks (Yin et al., 2023).

Model averaging and task arithmetic. Model merging is rooted in linear mode connectivity (Frankle et al., 2020), which posits that independently trained networks can lie in a shared low-error basin, enabling weight interpolation without catastrophic loss (Ainsworth et al., 2022). Model souping averages weights of models fine-tuned from a common checkpoint (Wortsman et al., 2022), and merging models from diverse training runs boosts out-of-distribution generalization (Rame et al., 2022). Task arithmetic (Ilharco et al., 2023) reframes fine-tuning as additive task vectors in nearly orthogonal directions, with TIES-Merging mitigating interference by pruning small updates and enforcing consensus signs (Yadav et al., 2023). Complementary approaches are based on the exclusion of a large proportion of delta entries from merging to confine task deltas (Yu et al., 2024; He et al., 2024), and compressing per-layer deltas via SVD with Procrustes alignment to reduce subspace overlap (Gargiulo et al., 2025).

Spectral analysis and model performance. RMT studies identify heavy-tailed self-regularization (HT-SR) as a key feature of well-trained models, where lower power-law exponents correlate with superior generalization in Transformers, serving as data-free quality predictors (Yang et al., 2022; Martin & Mahoney, 2019; 2021; Kothapalli et al., 2024). Training yields HT-SR shaped by the dynamics of the optimizer and consistently decreasing effective rank during training across architectures, with better generalizing solutions exhibiting lower effective rank (Yunis et al., 2024; Thamm et al., 2022; Staats et al., 2024). The Marchenko–Pastur (MP) law helps distinguish bulk eigenvectors—which are largely random—from the top singular components, which encode learned signal (Thamm et al., 2022; Staats et al., 2024).

6 Discussion and directions for further work

Our analysis allowed us to unveil several novel observations and to identify promising research directions: for pre-train, we establish complex spectral structure for matrices in attention blocks, which calls for developing a more complex theoretical model than the power-law tail extending from near-random bulk. Furthermore, since our analysis is based on the OLMo 2 model, extrapolating our findings to other architectures is an important future research direction.

We establish that domain adaptation during CPT occurs mostly via singular vectors direction changes, and those changes arise mainly for vectors associated with singular values near the peak of the spectrum. We hypothesize that complex spectral structure arising during pre-train allows the model weight matrices to accommodate more fine-grained changes and, accordingly, to better and faster adapt to new domains.

For CPT, we establish novel properties, resembling those of SFT deltas; in particular, domain connectivity, namely, the ability to interpolate after CPT on different domains, with interpolated model quality increasing with the increase in the pre-train token budget. Identifying the causes for this effect is an interesting research question. We further find that the CPT delta can be sparsified by up to *35%* over attention heads and *20%* over lowest singular values. This amount of sparsity, while not trivial, suggests that merging approaches such as DARE (Yu et al., 2024) would face significant challenges. Interestingly, relatively simple spectral metrics are poor predictors of an individual head's effect on model quality. This, coupled with the more complex spectral structures on later pre-train stages, calls for developing more sophisticated approaches to describing the spectral dynamics.

Overall, we expect that a combination of our framework for weight matrix analysis with the analysis of model activations can lead to significant advancements in model interpretability and presents an exciting avenue for future research.

REFERENCES

Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*, 2022.

C. Blakeney, M. Paul, B. W. Larsen, S. Owen, and J. Frankle. Does your data spark joy? performance gains from domain upsampling at the end of training. *arXiv preprint arXiv:2406.03476*, 2024. URL https://arxiv.org/abs/2406.03476.

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. URL https://arxiv.org/abs/1803.05457.

Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, November 2009. ISSN 1095-7200. doi: 10.1137/070710111. URL http://dx.doi.org/10.1137/070710111.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Hilton, Reiichiro Nakano, Jacob Hesse, John Schulman, Jared Kaplan, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The LLaMA 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pp. 3259–3269. PMLR, 2020.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodola. Task singular vectors: Reducing task interference in model merging. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18695–18705, 2025.

G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numer. Math.*, 14(5):403–420, April 1970. ISSN 0029-599X. doi: 10.1007/BF02163027. URL https://doi.org/10.1007/BF02163027.

Yuling Gu, Oyvind Tafjord, Bailey Kuehl, Dany Haddad, Jesse Dodge, and Hannaneh Hajishirzi. Olmes: A standard for language model evaluations, 2025. URL https://arxiv.org/abs/2406.08446.

Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats L. Richter, Quentin Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. Continual pre-training of large language models: How to (re)warm your model? *arXiv preprint arXiv:2308.04014*, 2023. URL https://arxiv.org/abs/2308.04014.

Yifei He, Yuzheng Hu, Yong Lin, Tong Zhang, and Han Zhao. Localize-and-stitch: Efficient model merging via sparse task arithmetic. *arXiv preprint arXiv:2408.13656*, 2024.

Truman Hickok. Scalable strategies for continual learning with replay. *arXiv preprint arXiv:2505.12512*, 2025. URL https://arxiv.org/abs/2505.12512.

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

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L. Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. Simple and scalable strategies to continually pre-train large language models. *arXiv preprint arXiv:2403.08763*, 2024. URL https://arxiv.org/abs/2403.08763.

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

Vignesh Kothapalli, Tianyu Pang, Shenyang Deng, Zongmin Liu, and Yaoqing Yang. Crafting heavy-tails in weight matrix spectrum without gradient noise. *arXiv preprint arXiv:2406.04657*, 2024.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Scott Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Charles H. Martin and Michael W. Mahoney. Traditional and heavy-tailed self regularization in neural network models. *arXiv preprint arXiv:1901.08276*, 2019. URL https://arxiv.org/abs/1901.08276.

Charles H. Martin and Michael W. Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021. URL https://jmlr.org/papers/v22/20-410.html.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

Zhenting Qi, Fan Nie, Alexandre Alahi, James Zou, Himabindu Lakkaraju, Yilun Du, Eric Xing, Sham Kakade, and Hanlin Zhang. Evolm: In search of lost language model training dynamics. *arXiv preprint*, 2025. Paper on language model training dynamics and continual pre-training.

Haoran Que, Jiaheng Liu, Ge Zhang, Chenchen Zhang, Xingwei Qu, Yinghao Ma, Feiyu Duan, Zhiqi Bai, Jiakai Wang, Yuanxing Zhang, et al. D-cpt law: Domain-specific continual pre-training scaling law for large language models. *Advances in Neural Information Processing Systems*, 37: 90318–90354, 2024.

Alexandre Rame, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 35:10821–10836, 2022.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019. URL https://arxiv.org/abs/1907.10641.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Pratyusha Sharma, Jordan T. Ash, and Dipendra Misra. The truth is in there: Improving reasoning in language models with layer-selective rank reduction. *ArXiv*, abs/2312.13558, 2023. URL https://arxiv.org/abs/2312.13558.

Max Staats, Matthias Thamm, and Bernd Rosenow. Locating information in large language models via random matrix theory. *arXiv e-prints*, pp. arXiv–2410, 2024.

Matthias Thamm, Bernd Rosenow, and Itamar Levi. Random matrix analysis of deep neural network weight matrices. *Physical Review E*, 106(5):054124, 2022. URL https://arxiv.org/abs/2203.14661.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *arXiv preprint arXiv:2302.00487*, 2023. URL https://arxiv.org/abs/2302.00487.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak 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 *International Conference on Machine Learning*, pp. 23965–23998. PMLR, 2022. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Yong Xie, Karan Aggarwal, and Aitzaz Ahmad. Efficient continual pre-training for building domain specific large language models. *arXiv preprint arXiv:2311.08545*, 2023. URL https://arxiv.org/abs/2311.08545.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115, 2023.

Yaoqing Yang, Ryan Theisen, Liam Hodgkinson, Joseph E Gonzalez, Kannan Ramchandran, Charles H Martin, and Michael W Mahoney. Evaluating natural language processing models with generalization metrics that do not need access to any training or testing data. *arXiv preprint arXiv:2202.02842*, 2022.

Lu Yin, Ajay Jaiswal, Shiwei Liu, Souvik Kundu, and Zhangyang Wang. Pruning small pre-trained weights irreversibly and monotonically impairs" difficult" downstream tasks in llms. *arXiv preprint arXiv:2310.02277*, 2023.

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 *Forty-first International Conference on Machine Learning*, 2024.

David Yunis, Kumar Kshitij Patel, Samuel Wheeler, Pedro Savarese, Gal Vardi, Karen Livescu, Michael Maire, and Matthew R Walter. Approaching deep learning through the spectral dynamics of weights. *arXiv preprint arXiv:2408.11804*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence?, 2019. URL https://arxiv.org/abs/1905.07830.

A TRAINING DETAILS

A.1 DATA

For pre-training, we use DCLM (Li et al., 2024) sample with 100B tokens in 4 mixes: 20B DCLM sample, 100B DCLM, 200B DCLM (oversampled), and 400B DCLM (oversampled).

For CPT we use data from FLAN decontaminated dataset, Dolmino High Quality Subset and Dolmino Math Mix, as proposed in OLMo 2 (OLMo et al., 2024), this data consists of language presented in the DCLM baseline. This is filtered by FastText and the FineWeb version of the original DCLM (Li et al., 2024). All mixes used in CPT are presented in Table 2

Table 1: Dataset composition for Dolmino High Quality Subset and Dolmino Math Mix.

Source	Type	Tokens	Words	Bytes	Docs
Mid-Training Dolmino High Quality Subset					
DCLM-Baseline *FastText top 7%* *FineWeb > 2*	High quality web	752B	670B	4.56T	606M
FLAN *from Dolma 1.7 decontaminated*	Instruction data	17.0B	14.4B	98.2B	57.3M
High quality total		**832.6B**	**739.8B**	**5.09T**	**710.8M**
Mid-Training Dolmino Math Mix					
TuluMath	Synthetic math	230M	222M	1.03B	220K
Dolmino SynthMath	Synthetic math	28.7M	35.1M	163M	725K
TinyGSM-MIND	Synthetic math	6.48B	5.68B	25.52B	17M
MathCoder2 Synthetic *Ajibwa-2023 M-A-P Matrix*	Synthetic math	3.87B	3.71B	18.4B	2.83M
Metamath *OWM-filtered*	Math	84.2M	76.6M	741M	383K
CodeSearchNet *OWM-filtered*	Code	1.78M	1.41M	29.8M	7.27K
GSM8K *Train split*	Math	2.74M	2.00M	25.3M	17.6K
Math total		**10.7B**	**9.73B**	**45.9B**	**21.37M**

Table 2: Continual pre-training mixes. * - oversampled data

Mix Name	Dolmino Math	DCLM Filtered	FLAN filtered
10B Mixes			
4BDM	4B	6B	-
8BDM	8B	2B	-
Math	10B	-	-
Text	-	10B	-
20B Mixes			
8BDM	8B	12B	-
16BDM	16B	4B	-
Math	20B*	-	-
Text	-	20B	-
50B Mixes			
20BDM	20B*	30B*	-
40BDM	40B*	10B	-
Math	50B*	-	-
Text	-	41.5B	8.5B

A.2 Models and Training Configuration

This study utilizes the OLMo 2 model architecture at two different scales: 1B. The architecture is based on the standard Transformer decoder Vaswani et al. (2023) and incorporates several modern enhancements:

- Removal of bias terms
- SwiGLU activation function
- Rotary positional embeddings (RoPE) with $\theta = 500,000$
- QKV clipping
- RMSNorm normalization
- Reordered layer norm (post-norm configuration)
- QK normalization
- Z-loss for training stability

All models are trained in mixed precision bfloat16. A complete description of the architecture and training methodology can be found in the original OLMo 2 paper OLMo et al. (2024).

Table 3: Model architecture and training hyperparameters.

	OLMo 2 1B
Model Architecture	
Hidden Dimension	2048
Number of Layers	16
Number of Attention Heads	16
MLP Ratio	8
Activation Function	SwiGLU
Normalization Type	RMS Norm
Positional Encoding	RoPE ($\theta = 500,000$)
Max Sequence Length	4096
Vocabulary Size	100,278
Training Configuration	
Global Batch Size	512

Pre-training. For our training setup, we adhere to the parameters proposed in the original OLMo 2 paper: a learning rate of 4×10^{-4} with a warmup phase over 0.7 billion tokens, followed by a cosine learning rate scheduler that decays to 10% of the initial rate by the end of training. The optimization is carried out using the AdamW optimizer Loshchilov & Hutter (2019) with the following hyper-parameters: β_1 = 0.9, β_2 = 0.95, and $\epsilon = 10^{-8}$. A weight decay of 0.1 is applied to all weights, including norms and biases, but not to embeddings.

Continual pre-training. The hyperparameters for continual pre-training remain consistent with those from pre-training, with the exception of the learning rate. In this stage, we start with the final learning rate from pre-training, which is $4 * 10^{-4}$, and apply a linear annealing schedule that decreases the learning rate to zero over the course of training.

B Reproducibility

We provide full source code to ensure all our experiments are reproducible. Our release includes the code for pre-training, CPT runs and commands for the OLMEs framework evaluation. The code is publicly available at: https://anonymous.4open.science/r/all-in-your-heads-CA28

We conduct a deep analysis of model weights using our open-source library, NetInspect https://anonymous.4open.science/r/netinspect-EF67

C DEFINITIONS

SINGULAR-VALUE TRANSPLANTATION

Given two checkpoints with per-layer weights $\boldsymbol{W}^{\text{ckpt1}} = \boldsymbol{U}_1 \boldsymbol{\Sigma}_1 \boldsymbol{V}_1^\top$ and $\boldsymbol{W}^{\text{ckpt2}} = \boldsymbol{U}_2 \boldsymbol{\Sigma}_2 \boldsymbol{V}_2^\top$ (SVD), the *transplanted* weight is

$$\widetilde{\boldsymbol{W}} = \boldsymbol{U}_2 \boldsymbol{\Sigma}_1 \boldsymbol{V}_2^\top. \tag{2}$$

Unless otherwise stated, transplantation is applied head-wise to Attention (Q, K, V, O) and MLP (FC1, Gate, FC2) weight matrices only; all other parameters remain from the target checkpoint.

SINGLE-HEAD REWIND

Let H be the number of heads and d_{head} the head dimension, so $d_{\text{model}} = H\, d_{\text{head}}$. We split projections along the head-concatenated axis and operate per head:

$$\boldsymbol{W}^Q = [\,\boldsymbol{W}^Q_{(0)} \mid \cdots \mid \boldsymbol{W}^Q_{(H-1)}\,],\ \boldsymbol{W}^K = [\,\boldsymbol{W}^K_{(0)} | \cdots | \boldsymbol{W}^K_{(H-1)}\,],\ \boldsymbol{W}^V = [\,\boldsymbol{W}^V_{(0)} | \cdots | \boldsymbol{W}^V_{(H-1)}\,], \tag{3}$$

where $\boldsymbol{W}^{Q/K/V}_{(i)} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{head}}}$ are column blocks. For the output projection $\boldsymbol{W}^O \in \mathbb{R}^{(H\, d_{\text{head}}) \times d_{\text{model}}}$, we split by rows into H blocks $\boldsymbol{W}^O_{(i)} \in \mathbb{R}^{d_{\text{head}} \times d_{\text{model}}}$ and reassemble by row concatenation. Single-head rewind at index i sets

$$\{\boldsymbol{W}^Q_{(i)}, \boldsymbol{W}^K_{(i)}, \boldsymbol{W}^V_{(i)}, \boldsymbol{W}^O_{(i)}\}^{\text{math}} \leftarrow \{\boldsymbol{W}^Q_{(i)}, \boldsymbol{W}^K_{(i)}, \boldsymbol{W}^V_{(i)}, \boldsymbol{W}^O_{(i)}\}^{\text{pre-train}}, \tag{4}$$

and replaces the corresponding QK-norm segments along the head axis with pre-train values.

AUC FOR REWIND CURVES

Let $\{k_t\}_{t=1}^T$ be the discrete percentages of heads rewound (in $[0, 100]$), and let $\mathcal{C}_\downarrow(k_t)$ and $\mathcal{C}_\uparrow(k_t)$ denote the metric at k_t for descending and ascending greedy orders, respectively. We first compute the discrete AUC of each curve using the trapezoidal rule:

$$\text{AUC}_\downarrow = \sum_{t=1}^{T-1} \frac{\mathcal{C}_\downarrow(k_t) + \mathcal{C}_\downarrow(k_{t+1})}{2} \big(k_{t+1} - k_t\big), \qquad \text{AUC}_\uparrow = \sum_{t=1}^{T-1} \frac{\mathcal{C}_\uparrow(k_t) + \mathcal{C}_\uparrow(k_{t+1})}{2} \big(k_{t+1} - k_t\big). \tag{5}$$

Our reported score is the absolute difference

$$\text{AUC-diff} \;=\; \big|\,\text{AUC}_\downarrow - \text{AUC}_\uparrow\,\big|. \tag{6}$$

When the grid is uniform, $k_{t+1} - k_t = \Delta$, this reduces to a constant Δ times the sum of trapezoid averages. Higher values indicate a greater separation between descending and ascending orders, while values near zero indicate random-like behavior.

DELTA TRUNCATION

Given $\boldsymbol{W} = \boldsymbol{U}\, \boldsymbol{\Sigma}\, \boldsymbol{V}^\top$ with $r = \min(m, n)$, setting $k = \lfloor (n/100)\, r \rfloor$ yields the top-k truncation

$$\widetilde{\boldsymbol{W}}_{(n\%)} = \boldsymbol{U}_{[:,1:k]}\, \boldsymbol{\Sigma}_{1:k,1:k}\, (\boldsymbol{V}^\top)_{[1:k,:]}, \tag{7}$$

where we keep the top-k singular directions and discard the rest. We report the kept fraction $n\%$.

D ADDITIONAL FIGURES ON MODEL-QUALITY METRICS

Included are figures for runs initialized from pre-train checkpoints at 20B, 100B, and 400B tokens, complementing the 4T case. These additions enable comparison across initial pre-train scales under identical continual pre-training setup, with model size held constant at 1B.

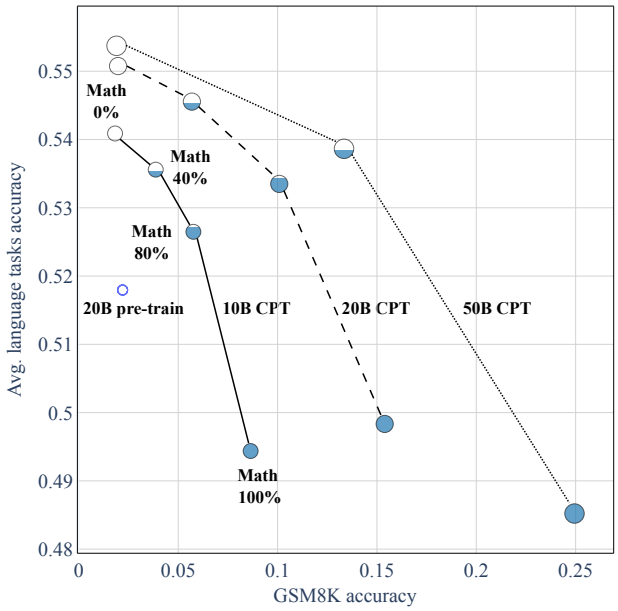

Figure 8: Quality of CPT models initialized from the 20B pre-train checkpoint, providing a detailed look at CPT configurations across token budgets and math proportions.

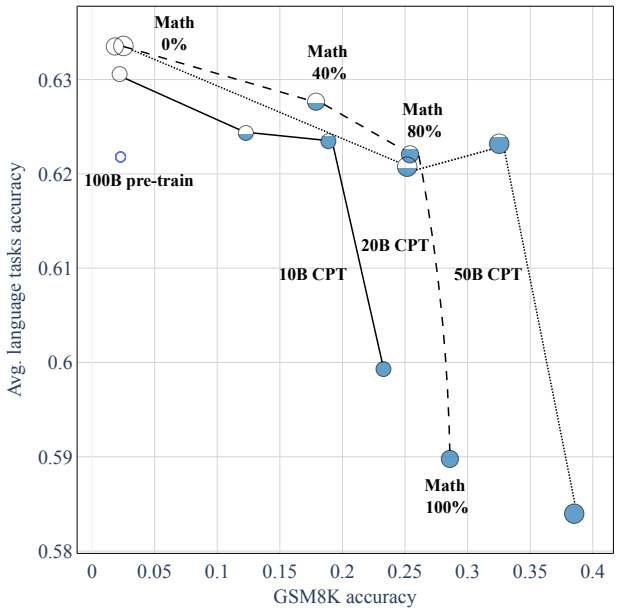

Figure 9: Quality of CPT models initialized from the 100B pre-train checkpoint, providing a detailed look at CPT configurations across token budgets and math proportions.

E CHECKPOINT MANIPULATIONS

Singular-vector adaptation. CPT updates are dominated by rotations of singular-vector subspaces, while singular value spectra remain nearly invariant. Layer-wise *singular-value transplantation* (for attention and MLP weights only) applies SVD to the pre-train checkpoint weights $\boldsymbol{W}^{\text{pre-train}} =$

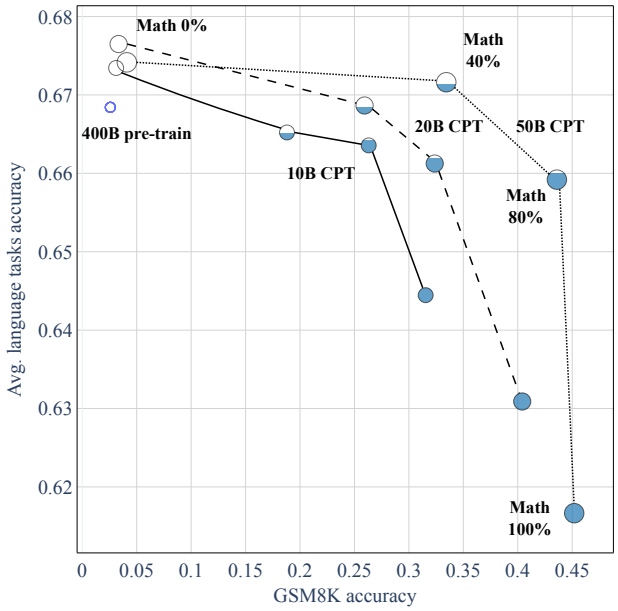

Figure 10: Quality of CPT models initialized from the 400B pre-train checkpoint, providing a detailed look at CPT configurations across token budgets and math proportions.

$\boldsymbol{U}^{\text{pre-train}}\,\boldsymbol{\Sigma}^{\text{pre-train}}\,(\boldsymbol{V}^{\text{pre-train}})^{\top}$ and the math checkpoint weights as $\boldsymbol{W}^{\text{math}} = \boldsymbol{U}^{\text{math}}\,\boldsymbol{\Sigma}^{\text{math}}\,(\boldsymbol{V}^{\text{math}})^{\top}$. We then form

$$\widetilde{\boldsymbol{W}} = \boldsymbol{U}^{\text{math}}\,\boldsymbol{\Sigma}^{\text{pre-train}}\,(\boldsymbol{V}^{\text{math}})^{\top}. \tag{8}$$

The results show that downstream performance metrics are preserved, indicating that domain knowledge acquired during CPT is encoded primarily in the geometry of singular vectors rather than in the singular value spectrum itself (see Fig. 11).

Supplementary tables

Table 4: Frobenius norm of the difference between $\boldsymbol{W}^{\text{text}}$ and $\boldsymbol{W}^{\text{math}}$ endpoints versus pre-train token budget. The distance decreases with longer pre-training.

Pre-train token budget	$\|\boldsymbol{W}^{\text{math}} - \boldsymbol{W}^{\text{text}}\|_F$
4T	1150
400B	1202
200B	1297
100B	1367
20B	1400
init	1535

Additional manipulations

(i) Injecting a small fraction (0.05–0.3) of the *text* delta into the 4T CPT *math* model does not improve Avg. language tasks accuracy and monotonically reduces GSM8K accuracy as the coefficient grows. (ii) "Super Mario"-style sparse deltas—formed by zeroing 60–90% of neurons in the 4T math delta and adding to pre-train—significantly degrade GSM8K accuracy. Finally, targeted SVD transplantation within MLPs shows that upper-tail singular vectors dominate GSM8K accuracy gains: we partition singular directions by magnitude as *upper* (top 25%), *bulk* (25–90%), and *lower* (bottom 10%); keeping CPT *upper* while using pre-train *lower/bulk* yields substantially

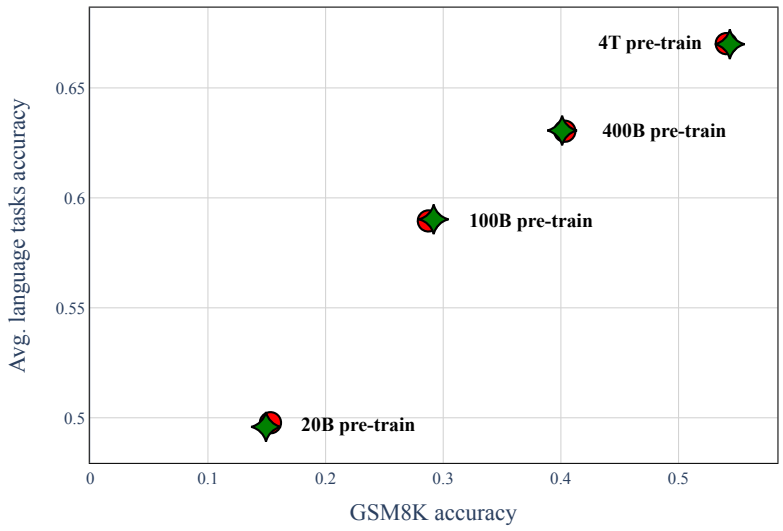

Figure 11: Singular-value transplantation results. Red circles denote CPT on DolminoMath, while star–diamond markers denote CPT on DolminoMath with singular values transplanted from the pre-train model. The results indicate that singular-value transplantation does not affect GSM8K accuracy.

Table 5: Scaling of interpolation quality versus joint training on an 80% math mixture as pre-train tokens increase. Relative increase is defined as (80% math mix GSM8K) divided by (80% math interpolation GSM8K).

Pre-train tokens	$W^{\text{interp}}(0.8)$ **(GSM8K accuracy)**	$W^{\text{mix}}(0.8)$ **(GSM8K accuracy)**	**Relative increase**
4T	0.436	0.463	+6%
400B	0.268	0.324	+21%
100B	0.188	0.257	+37%
20B	0.054	0.100	+85%

Table 6: Absolute area between descending and ascending rewind curves (AUC; higher is better). Smaller values indicate behavior closer to random, where ordering does not matter.

Heuristic	**AUC-diff**
greedy	19.1
overlap	5.2
delta Frobenius norm	5.2
delta stable rank	1.1
PL KS (pre-train)	0.9
random	0.0

higher GSM8K accuracy than the converse, and preserving only the top-10 singular directions from CPT is insufficient to recover the full improvement.

F NetInspect: Under the Hood

To systematically analyze the spectral characteristics and singular vector agreement adjustments in continual pre-training, we developed NetInspect, an open-source library designed for granular, architectural-component-level analysis of neural networks. The main text of the paper contains our key findings. This appendix provides a brief, practical overview of how to use our library to perform similar analyses.

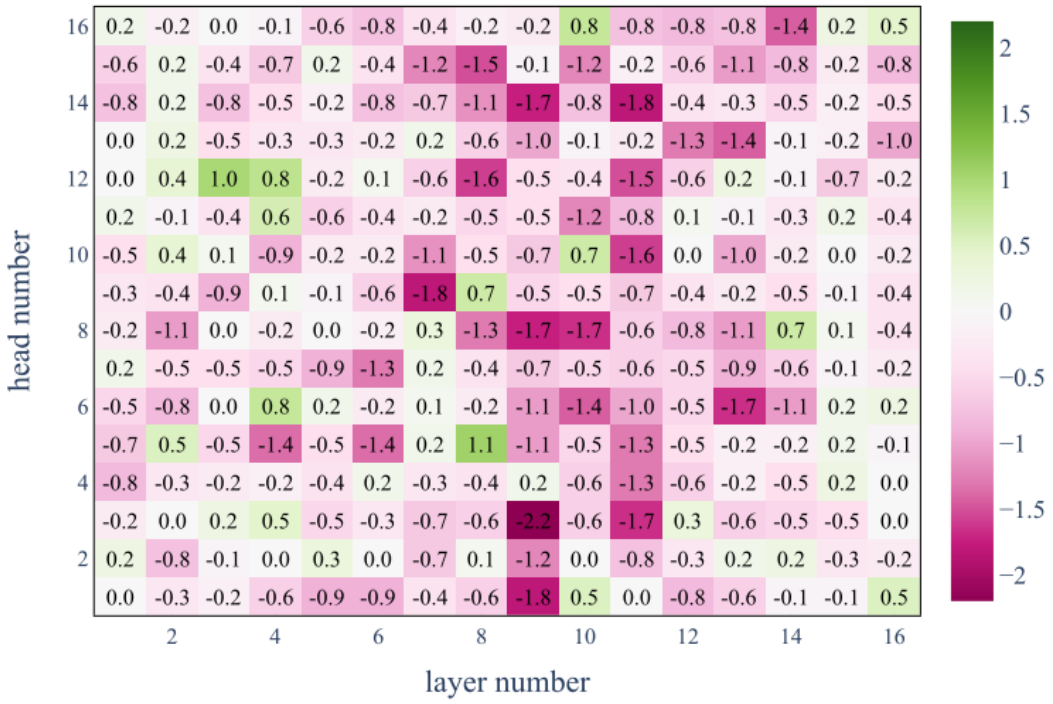

Figure 12: Head rewind heatmap (per-layer, per-head impact) for 20B Math CPT based on 4T pre-train. Cells contain GSM8K accuracy change (%).

Table 7: Targeted MLP transplantation ablations (moved from main text). Groups: upper = top 25%, bulk = 25–90%, lower = bottom 10%.

$W^{\text{pre-train}}$ components	W^{math} components	GSM8K accuracy
lower, bulk	upper	0.21
bulk, upper	lower	0.07
upper	lower, bulk	0.11
lower	bulk, upper	0.48
lower, bulk, upper	—	0.07
except top 10 leading	top 10 leading	0.07
—	lower, bulk, upper	0.53

F.1 Spectra-based LLM quality metrics

F.1.1 Norms

Our investigation includes the Frobenius ($\|\boldsymbol{W}\|_F$) and spectral ($\|\boldsymbol{W}\|_2$) norms:

$$\|\boldsymbol{W}\|_F = \sqrt{\sum_i \sigma_i(\boldsymbol{W})^2}, \tag{9}$$

$$\|\boldsymbol{W}\|_2 = \max_i \sigma_i(\boldsymbol{W}). \tag{10}$$

F.1.2 Ranks

We track the following structure-aware ranks:

Stable Rank:

$$R^s(\boldsymbol{W}) = \frac{\|\boldsymbol{W}\|_F^2}{\|\boldsymbol{W}\|_2^2}. \tag{11}$$

Effective Rank:

$$R^e(\boldsymbol{W}) = -\sum_{i=1}^{R(\boldsymbol{W})} \frac{\sigma_i(\boldsymbol{W})}{\sum_j \sigma_j(\boldsymbol{W})} \log\left(\frac{\sigma_i(\boldsymbol{W})}{\sum_j \sigma_j(\boldsymbol{W})}\right), \tag{12}$$

where hard rank $R(\boldsymbol{W})$ is the number of nonzero singular values (or a chosen truncation).

F.1.3 Singular Vector Agreement

Let $\boldsymbol{W}^i = \boldsymbol{U}^i\boldsymbol{\Sigma}^i\big(\boldsymbol{V}^i\big)^\top$ and $\boldsymbol{W}^j = \boldsymbol{U}^j\boldsymbol{\Sigma}^j\big(\boldsymbol{V}^j\big)^\top$. We summarize cosine similarities between left singular vectors via the *vector agreement* matrix

$$\boldsymbol{A}^u(\boldsymbol{W}^i, \boldsymbol{W}^j) = |\big(\boldsymbol{U}^i\big)^\top \boldsymbol{U}^j|, \tag{13}$$

where the absolute value is taken element-wise. We report two levels: per-vector (diagonal and row-maximum agreements) and a global average given by the mean of diagonal agreements across the matrix.

F.1.4 Random Matrix Theory Fits

The HT-SR (Heavy-Tailed Self-Regularization) theory originally emerged as a semi-empirical theory, and early seminal works (Martin & Mahoney, 2019; 2021) studied the empirical spectral density of weight matrices given by

$$\mu(\lambda; \boldsymbol{X}^i) = \frac{1}{n}\sum_{j=1}^{n} \delta\left(\lambda - \lambda_j(\boldsymbol{X}^i)\right). \tag{14}$$

Here $\boldsymbol{X}^i$ is a correlation matrix, defined as $\boldsymbol{X}^i = \frac{1}{m}\big(\boldsymbol{W}^i\big)^\top \boldsymbol{W}^i$. The eigenvalues of $\boldsymbol{X}^i$ provide insight into the distribution of information across different directions in the feature space. This is captured by the Empirical Spectral Density (ESD),

where $\lambda_1(\boldsymbol{X}^i) \leq \ldots \leq \lambda_n(\boldsymbol{X}^i)$ are the eigenvalues of $\boldsymbol{X}^i$, and δ denotes the Dirac delta function. The ESD thus represents a probability measure describing how the eigenvalues are distributed. Note the relation between singular values and correlation eigenvalues: $\lambda_i(\boldsymbol{X}) = \frac{1}{m}\,\sigma_i(\boldsymbol{W})^2$.

At random initialization weights are modeled as entries drawn from a Gaussian Orthogonal Ensemble (GOE)

$$W_{i,j} \sim \mathcal{N}(\boldsymbol{x}; 0, \mathrm{Var}(\boldsymbol{W})). \tag{15}$$

One of the key observations in modern DNNs is the deviation of ESDs from classical RMT predictions for such matrices, such as the Marchenko–Pastur (MP) distribution

$$\rho^{\mathrm{MP}}(\lambda; \boldsymbol{X}) = \begin{cases} \dfrac{n}{2\pi m \cdot \mathrm{Var}(\boldsymbol{W})} \dfrac{\sqrt{(\lambda^{\max} - \lambda)(\lambda - \lambda^{\min})}}{\lambda}, & \lambda \in [\lambda^{\min}, \lambda^{\max}], \\ 0, & \text{otherwise}, \end{cases} \tag{16}$$

with

$$\lambda^{\max\,/\,\min}(\boldsymbol{X}) = \mathrm{Var}(\boldsymbol{W})\big(1 \pm \sqrt{n/m}\big)^2. \tag{17}$$

While random GOE matrices follow the classical MP distribution, trained neural network weight matrices deviate significantly from this behavior (Martin & Mahoney, 2019; 2021). During training, a non-random (signal) component typically emerges outside the MP bulk. We delineate bulk versus outliers by estimating an upper edge $\widehat{\lambda}^{\max}$ via a rescaled MP heuristic (Martin & Mahoney, 2021). To be more precise, for a particular matrix $\boldsymbol{W}$ we:

1. Randomise $\boldsymbol{W}$ by permuting its elements.
2. Compute the empirical scale $s(\boldsymbol{W})$ from the randomized matrix.
3. Find $\lambda^{\max}$ based on the MP prediction with $s(\boldsymbol{W})$.

4. Correct the scale

$$\widehat{s}^2 = s(\boldsymbol{W})^2 - \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{\{\lambda_i > \lambda^{\max}\}}\, \lambda_i, \tag{18}$$

then find $\widehat{\lambda}^{\max}$ based on $\widehat{s}$.

Well-trained models further develop heavy tails with $\rho^{\mathrm{PL}}(\lambda; \boldsymbol{X}) = c \cdot \lambda^{-\alpha}$ with normalization constant c and scaling exponent $\alpha \in (1.5, 5)$ (Clauset et al., 2009); smaller α indicates stronger correlations and, empirically, higher model quality (Martin & Mahoney, 2021).

We estimate α by Maximum Likelihood Estimation (MLE) estimator (Clauset et al., 2009); empirical evidence shows MLE performs well for $\alpha \in [1.5, 3.5]$ (Martin & Mahoney, 2021), a typical range for DNN weight matrices. Fit quality is checked with the Kolmogorov–Smirnov (KS) distance between the empirical spectrum CDF $S(\lambda)$ and the fitted PL CDF $\widehat{S}(\lambda)$

$$\mathrm{KS} = \sup_{\lambda \in \{\lambda_i(\boldsymbol{W})\}} \left| S(\lambda) - \widehat{S}(\lambda) \right|. \tag{19}$$

F.2 Visualizations

The pipeline consists of five sequential stages, each designed to probe different aspects of the network's weight matrices. We assume the user is comparing two model checkpoints, $\boldsymbol{W}^{\text{ckpt1}}$ and $\boldsymbol{W}^{\text{ckpt2}}$, and checkpoint of their delta ($\boldsymbol{W}^{\text{ckpt1}} - \boldsymbol{W}^{\text{ckpt1}}$).

Box plots: Component wise spectral metrics distribution

The analysis employs comparative box plots generated for all weight matrices, which are organized by matrix family—specifically, the Attention projections ($\boldsymbol{W}^{\mathrm{Q}}$, $\boldsymbol{W}^{\mathrm{K}}$, $\boldsymbol{W}^{\mathrm{V}}$, $\boldsymbol{W}^{\mathrm{O}}$) and the MLP layers. To ensure a granular analysis, each attention matrix is first decomposed into its heads before any metric computation.

The analysis itself is structured by distinct metric categories for precise comparison: one group focuses on norms, including the Frobenius and Spectral norm, while another group concentrates on rank measures, namely the Stable rank and Effective rank. For the MLP matrices exclusively, this set of metrics is augmented with the Kolmogorov-Smirnov (KS) statistic, which quantifies the fit to both a Power Law (PL) and a Marchenko-Pastur (MP) distribution.

The visual encoding of the box plots is designed to convey multiple data dimensions simultaneously. The fill color of each box plot signifies the specific matrix parameter. The outline color denotes the checkpoint, where blue represents ckpt1, red represents ckpt2, and green represents the delta. Furthermore, a jittered scatter plot is laid near each box; the individual data points are colored on a continuous spectral scale from blue to red. This color mapping corresponds directly to the layer index, with blue indicating layers near the model input and red indicating layers near the output. This integrated approach allows for the immediate assessment of distributional properties—including medians, quartiles, and outliers—across the two checkpoints, while preserving the crucial ability to discern depth-dependent patterns within the metric distributions.

Cluster bar charts: Singular values distribution

The purpose of this stage is to visualize the entire distribution of singular values. The visualization employs a cluster bar chart where the x-axis is exponential binning of the singular values, while the y-axis represents the count of values falling into each bin.

To articulate the layer-by-layer evolution of the spectrum, the color of each bar corresponds to its layer index. For attention matrices, the singular values are first averaged across all heads within a given layer before the binning process.

Sparklines: Spectral metric trends

This stage aims to compactly visualize the trend of each metric across the network's depth, enabling a quick, at-a-glance assessment of layer-wise patterns. The procedure involves plotting the metric

value as a function of layer index for each matrix family. For attention matrices, the values are aggregated across their heads: a solid line represents the mean, a dashed line represents the median, and a shaded area delineates the band of mean ± one standard deviation. These resulting sparklines provide a temporal view of metric evolution through the network's layers. This visualization allows for the immediate identification of critical patterns, where a sharp change in a metric at a specific layer or a consistent divergence between the mean and median can signal important architectural transitions or anomalies.

HEATMAPS: SPECTRAL METRIC HETEROGENEITY

The purpose of this stage is to expose fine-grained, head-specific and layer-specific variations in metrics, providing a direct visual comparison of the differences between the two checkpoints. The procedure involves creating a two-dimensional grid for each combination of attention matrix type ($\boldsymbol{W}^{\mathrm{Q}}$, $\boldsymbol{W}^{\mathrm{K}}$, $\boldsymbol{W}^{\mathrm{V}}$, $\boldsymbol{W}^{\mathrm{O}}$) and metric, with the axes representing the layer index and head index. For each such combination, a quartet of heatmaps is generated: the first visualizes the raw metric values for $\boldsymbol{W}^{\mathrm{ckpt1}}$ on a blue color scale; the second visualizes the raw values for $\boldsymbol{W}^{\mathrm{ckpt2}}$ on a red scale; the third shows the delta and the fourth displays a metric difference. This visualization allows for pinpointing the exact heads and layers that contribute most significantly to the overall divergence between the two models, moving from a high-level summary to a precise diagnostic tool.

VIOLIN PLOTS: SINGULAR VALUE DISTRIBUTION AND SINGULAR VECTOR AGREEMENT

The final stage of our analysis is architected the most detailed view of the singular value distribution for each individual head and layer, while simultaneously incorporating information about singular vector agreement. This is achieved by generating a matrix of plots, where each cell corresponds to a specific (layer, head) pair. Within a given cell, a violin plot is used to depict the density of the singular values for that specific matrix. Overlaid onto this violin plot is a jittered scatter plot, where the color of each point is determined by a vector agreement metric—overlap between the corresponding singular vectors of $\boldsymbol{W}^{\mathrm{ckpt1}}$ and $\boldsymbol{W}^{\mathrm{ckpt2}}$. This powerful composite visualization synergistically combines the shape of the singular value spectrum, conveyed by the violin, with the stability of the underlying directional components, indicated by the colored jitter, across the entire model. It thereby enables the precise identification of nuanced scenarios, such as attention heads that exhibit a similar spectral distribution but possess different singular vectors directions, or vice versa, offering insight into the micro-dynamics of network evolution.