Understanding Gradient Orthogonalization Effects in Deep Ensembles through the Topology of Parameter Trajectories and Activations

Matthew O'Malley-Nichols
University of California, San Diego
momalleynichols@ucsd.edu

Abstract

We present the topological analysis of gradient orthogonalization effects in neural network optimization. Using persistence landscapes derived from Vietoris-Rips filtrations of activation spaces, we characterize the geometric signatures of seven optimization algorithms including Muon, Adam, and spectral filtration variants. Additionally, we explore the effects of spectral filtering on the exploration of parameter configurations. We observe that generalization requires balanced topological complexity in learned representations.

1 Introduction

Deep ensembles provide a functionally diverse approximation of the Bayesian model average, giving uncertainty estimates over modes in the loss manifold (Wilson and Izmailov, 2022). Gradient-orthogonalization algorithms have empirically been shown to improve convergence speed and stability (Jordan, 2024) by decorrelating successive updates. With an emergence of work exploring spectral dynamics ((Yang et al., 2024), (Doikov et al., 2024)) and functional smoothness (Belkin, 2021) for generalization, we hypothesize that gradient-orthogonalization methods will improve deep ensembles through increases in parameter diversity. We hope to identify any ties between topological invariants in our considered spaces and the improved generalization prevalent in smooth interpolants.

2 Previous Works

2.1 Gradient Orthogonalization

(Oymak and Soltanolkotabi, 2018) notice that first-order stochastic optimization methods converge to the nearest global optima, regardless of the supplied learning rate. (Ghorbani et al., 2019) show that non-batch normalized networks possess a heavy-tailed Hessian eigen-spectrum, which can slow convergence speed through eigenvalue outliers. (Huang et al., 2020) impose orthogonal penalties on their loss functions to maintain near-orthogonal gradients, while (Gupta et al., 2018)'s Shampoo maintains left and right preconditioner matrices updated by a running average of the gradient's covariance along each dimension. (Bernstein and Newhouse, 2024) propose to replace the Euclidean norm used in the inner product for the step size measurement with a data-driven matrix norm. This data-driven matrix norm can orthogonalize collinear updates in the new geometry. (Feng et al., 2025) prove that keeping the spectrum of the layer-wise Jacobians within a narrow band (by spectral clipping or orthogonalization) shrinks the global Lipschitz constant and improves the weight matrices condition numbers. The Muon optimizer (Jordan, 2024) performs momentum on the gradient update, then orthogonalizes it with Newton-Schulz-5. The orthogonalization removes components along dominant gradient directions, leading to empirical speedups in training times, such as a world record on the NanoGPT benchmark in 2024 (Jordan, 2024). (Liu et al., 2025) scales Muon for large language models, showing faster convergence and smoother loss curves than AdamW.

2.2 Generalization in Neural Networks

(Belkin, 2021)'s interpolation framework links benign overfitting to the smoothness of the underlying interpolant, promoting low-frequency function representations. A Fourier analysis approach by (Rahaman et al., 2019) examines the tendency for networks to learn low frequencies first, in a phenomenon they name the spectral bias. (Canatar et al., 2021) find the alignment of the target function's spectrum and the Neural Tangent Kernel's (NTK) eigen-basis to govern the spectral bias phenomenon, implying it is caused by task-model spectral alignment. (Fridovich-Keil et al., 2022) find the same dynamics in the empirical NTK for finite-width models. (Yang et al., 2024) connects the feature learning (rich) and lazy regimes for neural

networks based on a spectral-scaling law dictating when spectral bias can emerge. (Yunis et al., 2024) find a correlation between linear mode connectivity in the weight space and the agreement of top singular values.

While generalization phenomena such as double descent can be examined through the spectral dynamics of weights, (Wilson and Izmailov, 2022) explains these phenomena by modeling a multimodal posterior. In contrast to the previous chains of thought, they view the optimization dynamics simply as a proxy for which modes are reachable, and view robust generalization as a result of integrating over diverse modes.

2.3 Modal Connectivity

The manifold hypothesis postulates the existence of a low-dimensional manifold in the parameter space upon which low-loss hypotheses exist (Belkin, 2021). (Buchanan et al., 2021) support this idea by showing that independently trained models can be connected by continous, low-loss curves. (Şimşek et al., 2021) show how symmetric group actions (such as permutations, sign flips, and scalings) create flat minima manifolds in the parameter space. (Benton et al., 2021) and (Wortsman et al., 2021) generalize the linear mode connector to higher order simplexes. (Chen and Saidi, 2025) compares these methods with a non-linear subspace parameterized by a MLP, finding the methods performed equally. (Mao et al., 2024) find a task-dependent low-dimensional manifold in parameter space connected by a geodesic metric based on empirical Fisher. Additionally, they find that projecting test-set gradients onto the manifold's orthogonal complement annihilates them, implying that learning depends on the model's position along the manifold. (Huh et al., 2024) theorize the low dimensional manifold is also shared for multi-modal learning problems.

2.4 MultiSWAG

MultiSWAG (Wilson and Izmailov, 2022) extends the Stochastic Weight Averaging-Gaussian (SWAG) (Maddox et al., 2019) framework to approximate multi-modal posterior distributions in neural networks. While SWAG fits a single Gaussian to the trajectory of stochastic gradient descent iterates (intra-model weight snapshots) for a single model, MultiSWAG trains multiple SWAG models with diverse training trajectories in order to converge to distinct local minima. Each of these SWAG models acts as a Monte Carlo particle, providing an estimate of a mode of the posterior. Each particle is trained normally for a pretraining phase, then in the SWAG phase the mean and covariance of the weights for each particle are computed using the following updates: Let $\mathbf{w}_i$ denote the weights at the i-th collected SGD iterate, and let T be the total number of iterates. The SWAG mean is given by

$$\mathbf{w}_{\mathrm{SWA}} = \frac{1}{T}\sum_{i=1}^{T}\mathbf{w}_i, \tag{1}$$

while the covariance is estimated as a sum of a low-rank and diagonal component, both formed by the deviation between swag mean and particle mean:

$$\Sigma = \frac{1}{T-1}\sum_{i=1}^{T}(\mathbf{w}_i - \mathbf{w}_{\mathrm{SWA}})(\mathbf{w}_i - \mathbf{w}_{\mathrm{SWA}})^{\top} + \mathrm{diag}\left(\frac{1}{T}\sum_{i=1}^{T}(\mathbf{w}_i - \mathbf{w}_{\mathrm{SWA}})^2\right) \tag{2}$$

Each SWAG particle thus provides a Gaussian approximation $\mathcal{N}(\mathbf{w}_{\mathrm{SWA}}, \Sigma)$ for its local mode. At inference time, MultiSWAG samples weights the mixture of Gaussians formed by the particles and averages predictions, effectively performing Bayesian model averaging across multiple basins of attraction. This approach enables MultiSWAG to capture a multi-modal posterior, leading to improved uncertainty quantification and predictive performance (Wilson and Izmailov, 2022).

2.5 Topological Data Analysis

2.5.1 Vietoris–Rips Complex

Dey and Wang define the Vietoris–Rips complex of a finite metric space (P, d) at scale r as to be the complex with pairwise vertex distances bounded by $2r$:

$$\mathbb{VR}^r(P) = \big\{\sigma \subset P : d(p,q) \le 2r \;\forall\, p,q \in \sigma\big\},$$

and show it interleaves with the tighter Čech complex via $\mathbb{C}^r(P) \subset \mathbb{VR}^r(P) \subset \mathbb{C}^{2r}(P)$ (Dey and Wang, 2022, Def. 2.10, Prop. 2.2). In Euclidean spaces the 1-skeletons (graphs) coincide (Dey and Wang, 2022, Fact 2.2) when metric balls $B(p,r)$ and $B(q,r)$ are not disjoint, ensuring that the Vietoris-Rips captures the same connectivity information as the Čech.

The Vietoris-Rips filtration is the collection of Vietoris-Rips complexes formed by the simplicial

inclusion map induced by increasing the scale parameter r (Dey and Wang, 2022, Def. 3.1):

$$\mathbb{VR}(P) = \{\mathbb{VR}^r(P)\}_{r\in[0,\infty)},$$

2.5.2 Persistent Homology and Stability

Persistent homology tracks the birth and death scales of topological features across a filtration of spaces $\{K_r\}$. A persistence interval $[b_i, d_i)$ denotes birth and death scales of homology group features. The persistence diagram Dgm_p corresponds to the multiset of points $\mathrm{Dgm}_p = \{(b_i, d_i)\} \subset \mathbb{R}^2$. For simplex-wise monotone functions $f, g : K \to \mathbb{R}$, the Stability Theorem for simplicial filtrations bounds the bottleneck distance d_b in the space of persistence diagrams by:

$$d_b\big(\mathrm{Dgm}_p(\mathcal{F}_f),\, \mathrm{Dgm}_p(\mathcal{F}_g)\big) \;\le\; \|f - g\|_\infty$$

(Dey and Wang, 2022, Thm 3.2, Thm 3.3). Thus, small perturbations in the data or filter induce at most that magnitude of infinity-norm change.

2.5.3 Betti Curves

The p-th Betti curve counts the number of active features at a filtration scale r:

$$\begin{aligned}\beta_p(r) &= \operatorname{rank} H_p(K_r) \\ &= \#\{(b_i, d_i) \in \mathrm{Dgm}_p : b_i \le r < d_i\}\end{aligned}$$

The Betti curve offers a summary of connected components ($p = 0$) and loops ($p = 1$) as functions of r (giotto-tda contributors, 2023).

2.5.4 Persistence Landscapes

Persistence landscapes embed persistent diagrams into a Banach Space $\mathcal{L}^p(\mathbb{N} \times \mathbb{R})$ via the fucntion $\lambda_D : \mathbb{N} \times \mathbb{R} \to \mathbb{R}$. As the p-norm is defined on the space, the p-landscape distance is defined in (Dey and Wang, 2022, Def. 13.1) by:

$$\Lambda_p(\mathrm{Dgm}_1, \mathrm{Dgm}_2) = \left\|\lambda_{\mathrm{Dgm}_1} - \lambda_{\mathrm{Dgm}_2}\right\|_p$$

When the $p = 2$, the space $\mathcal{L}^2$ is a Hilbert space with stable $\mathcal{L}^2$ distance, allowing for statistical operations such as the mean between diagrams.

2.5.5 Mapper and Topological Graph Methods

Mapper constructs a simplicial nerve over an overlapping cover induced by a lens function $f : X \to \mathbb{R}^m$. One covers $f(X) \subset \mathbb{R}^m$ by overlapping bins $\mathcal{U} \in \{U_i\}$, clusters each preimage $f^{-1}(U_i) \subset X$, and builds the nerve by connecting clusters with nonempty overlap. By the Nerve Theorem, the nerve reflects the homotopy of the union of pullbacks when intersections are contractible (Singh et al., 2007) .

2.6 Our Work

While the spectral bias perspective and multimodal posterior view might seem disconnected, our work positions these frameworks as complementary rather than competing. Gradient orthogonalization methods like Muon (Jordan, 2024) modify the spectral properties of optimization trajectories, potentially affecting both the spectral characteristics of learned functions and the diversity of discovered modes in parameter space.

We hypothesize that by altering the spectral properties of gradient updates, orthogonalization methods simultaneously affect the smoothness of the interpolant and the diversity of the modes discovered in parameter space. This connection is similar to the results found by (Chen and Saidi, 2025), who demonstrated that linear mode connectivity correlates with agreement in top singular values between models. Gradient orthogonalization, by modifying the spectral properties of updates, could therefore impact both the functional smoothness and modal diversity, which can be uncovered by a topological analysis of the corresponding spaces.

3 Methodology

3.1 MultiSWAG Ensemble

We follow the MultiSWAG protocol (Maddox et al., 2019) to approximate a multi-modal posterior over network parameters. For each optimizer described below, we train an ensemble of $N = 20$ particles initializing with the same parameters for 30 pretraining epochs and 20 SWAG epochs. We bootstrap our training data to simulate independent training runs from the same initialization. At inference time, we draw 10 samples per particle from each particle's Gaussian approximation and average their predictions, yielding a Bayesian model average over 200 weight samples.

3.2 Optimizers

We compare the following seven optimization strategies:

- **Adam** (Kingma and Ba, 2017): the standard adaptive moment estimator.

- **AdamW** (Loshchilov and Hutter, 2019): Adam with decoupled weight decay.
- **Muon** (Liu et al., 2025): applies a Newton–Schulz orthogonalization step to the momentum update.
- **Muon10p** & **10p**: Muon and AdamW with only 10% of gradient singular values maintained, and the rest zeroed out.
- **MuonSpectralNorm** & **SpectralNorm**: Muon and AdamW with only the top gradient singular value maintained, and the rest zeroed out.

The ablations isolate the effects of spectral filtering on both optimization dynamics and topological signatures.

3.3 Parameter PCA

For computational efficiency of our parameter space analyses, we perform a PCA analysis on the final pretraining parameters of particle 0 for each optimizer. We flatten the weights to $\mathbb{R}^D$ and compute the principal components, analyzing the variance explained. We observe the following variance explained for 50 principle components:

Optimizer	Variance Explained (%)
Muon10p	96.2
MuonSpectralNorm	95.4
AdamW	94.3
10p	94.3
Adam	64.4
Muon	92.3

Table 1: Fraction of variance captured by the top 50 principal components of the final pretraining weights (particle 0).

With over 90% variance explained for all except the Adam optimizer, we use the top-50 principle components of the flattened parameters for our parameter space analyses.

3.4 Trajectory Collection

For each epoch of the pretraining phase for each particle, we save model parameters, training and validation loss, validation and validation corrupt accuracy, and mean and standard deviation of epoch gradients' L^2 and spectral norms for each layer. For each epoch of the SWAG phase for each particle, we collect the accuracies using particle means and save first and second moments of parameters as well as square root low rank covariance approximations.

4 Experiments

4.1 Model

We use a fully-connected network with 2 hidden layers and ReLU nonlinearity for each of our MultiSWAG particles. The model architecture can be seen in 2 below.

Layer	Input Size	Output Size
FCN + ReLU	784	256
FCN + ReLU	256	256
FCN	256	10

Table 2: Architecture of the 3-layer MLP

4.2 Datasets

We use the MNIST dataset, comprised of 60k training images across 10 classes of 28x28 grayscale handwritten digits. Images are preprocessed by normalizing to mean 0.1307 and standard deviation 0.3081, then flattened to 784-dimensional vectors. For validation, we use a holdout set of 10k samples from the training set. Additionally, we create a corrupted dataset we denote MNIST-C by performing: apply random affine transformations with 20-degree rotation, translation of (0.1, 0.1), and scaling factors between (0.9, 1.1), add Gaussian noise $\mathcal{N}(0, 0.2)$, then normalize and flatten.

4.3 Training Configuration

We use a batch size of 128, a learning rate of $\eta = 1e-2$, and a weight decay of $\lambda = \frac{1e-2}{1e3} = 1e-5$ for each optimizer. For the AdamW variants we use $\beta = (0.9, 0.95)$ and $\epsilon = 1e{-}10$. For the Muon layers we use $\mu = 0.95$ momentum and 10 Newton Schulz steps.

4.4 Parameter Space Analysis

We analyze parameter trajectory geometry using Mapper (Singh et al., 2007) applied to weight trajectory point clouds $\{\mathbf{w}_e^{(i)}\}_{e,i}$ built using the first 50 principle components of the particle parameters. We use a UMAP projection to two components for our lens, noting a minimum trustworthiness (Venna and Kaski, 2001) score of 0.995 with

UMAP compared to 0.75 with the first two PCA components. After a qualitative hyper-parameter search between $c \in \{1, 2, 4, 5, 8, 12\}$ number of cubes and $p \in \{0.1, 0.2, 0.4, 0.5\}$ percent overlap, we decide to use 5 cubes and 50% overlap. Within each cover element, we perform k-means clustering ($k = 3$) and construct the nerve complex, yielding Mapper graphs that reveal parameter space connectivity and mode structure for the trajectories.

4.5 Activation Space Topological Analysis

We extract penultimate-layer activations $\mathbf{A}_e^{(i)} \in \mathbb{R}^{128\times256}$ at epochs $e \in \{0, 30\}$ for each particle i on a batch of validation and corrupted validation data. For each activation matrix $\mathbf{A}_e^{(i)} \in \mathbb{R}^{128\times256}$, we construct Vietoris-Rips complexes using GUDHI and perform filtrations with an adaptive maximum radius set to the 90th percentile of pairwise Euclidean distances. We process the resulting persistence diagrams through three specialized analysis pipelines.

- `analyze_vietoris_rips.py` computes Betti curves $\beta_p(\epsilon)$ by evaluating feature counts at $1,000$ linearly spaced filtration radii, constructs persistence landscapes via tent functions sampled at $1,000$ t-points, and performs Mann–Whitney U-tests on H_1 feature counts as well as bootstrap 95% confidence intervals using $1,000$ resamples. It outputs the multi-panel Betti-curve plots and landscape-distance heatmaps to compare topological similarity across optimizers.

- `analyze_persistence_diagrams.py` extracts per-diagram statistics including total and finite feature counts, lifetime summaries (mean, standard deviation, maximum), and persistence entropy (Myers et al., 2019):

$$E(\mathrm{Dgm}) = -\sum_i p_i \log p_i$$
$$p_i = \frac{d_i - b_i}{\sum_j (d_j - b_j)}$$

  It performs epoch-comparison analyses (0 vs. 30) to measure changes in feature counts and lifetimes, and cross-optimizer comparisons that generate heatmaps of mean feature counts and maximum lifetimes. Bottleneck distances are computed via the `persim` library.

Figure 1: Validation Accuracy over Epochs

- `analyze_landscapes.py` builds a persistence-landscape database by sampling each landscape at 500 t-points. It then performs:

  - *Time-series tracking:* computes $\|\Lambda_i^{(0)} - \Lambda_i^{(30)}\|_2$ for each particle i to quantify topological drift.
  - *Stability analysis:* measures inter-particle consistency by computing mean pairwise $\mathcal{L}^2$ landscape distances within each optimizer ensemble for a measurement of topological diversity.

  The script generates box plots of drift distances and stability-score visualizations for each optimizer.

5 Results

We provide CSV files and plots for our results in the `/results/` folder of the GitHub Repository. A majority of the plots are also available in the Appendix 6.

5.1 Pretrain Accuracy

1 provides validation accuracy per epoch for both the validation and validation corrupted datasets. The Muon and Muon10p variant outperform all other optimizers on both datasets. MuonSpectralNorm performs similarly to the other optimizers on the clean dataset, but has by far the lowest accuracy on the corrupt dataset.

5.2 Gradient Norms

We see in 4, 6, and 5 that the default Muon optimizer's total $\mathcal{L}^2$ gradient norm variation increases during training. Additionally, the Muon10p's gradient norm coefficient of variation skyrockets past epoch 10, though it still attains the highest accuracy. As the relative variation increases, the optimizer's mean gradient norm still decreases. This implies that the average step sizes are globally shrinking leading to stable weights, though some low rank directions are being amplified. This may possibly lead to the optimizer escaping narrow basins, explaining the high validation accuracy of the Muon10p model.

5.3 SWAG Covariance Matrices

In the SWAG training phase, the covariance matrix of Muon optimizer particles display a larger log-determinant 8 but smaller trace 9 and eigenvalue coefficient of variation 10 than the other optimizers. This implies the Muon optimizer induces a posterior distribution with significant uncertainty across many parameter directions, without contracting its spread of doubt. This supports our initial hypothesis that the Muon optimizer would explore a larger portion of parameter space. Additionally, the high posterior diversity provides more effective modes for Bayesian marginalization, reducing approximation error through a rich set of hypotheses.

5.4 Mapper of Parameter Trajectories

We provide all the UMAP Mapper graphs in E. The default Muon parameter trajectory Mapper graph 15 displays a higher node count than Adamw 14, indicating more exploration of parameter space. Compared to Muon, the spectral filtered Muon10p 17 maintains the exploration while reducing connectivity, creating chain-like sequences of minimally overlapping clusters. In contrast, MuonSpectralNorm 18 drastically increases connectivity ($175 \rightarrow 223$ edges) without affecting the node count, indicating exploration in narrow subspaces.

The connectivity of the parameter trajectory Mapper graphs inversely relate to the validation accuracy and gradient coefficient of variation trends. Sparser Mapper connectivity, indicating exploration along broad subspaces, aligns with better downstream task performance and less erratic updates.

5.5 Activation Landscapes

MuonSpectralNorm exhibits the highest mean interparticle landscape distance (≈ 0.200) 21, reflecting its variable action-space topology. Muon10p presents a more moderate distance (≈ 0.025), while the other optimizer ensembles have little variation in activation topology. The validation performance of Muon10p and MuonSpectralNorm underscore that while moderate increases in functional topology diversity can increase performance, excessive increases in diversity reduce performance. Interestingly, corrupted data increases the landscape distance between Muon and Muon10p but decreases the landscape distance between Muon10p and MuonSpectralNorm F.3.

The landscape drift plots F.2 reveal the Adam optimizer variants lead to larger $\mathcal{L}^2$ activation landscape shifts than Muon. The Muon10p and MuonSpectralNorm activation landscapes move significantly less than the Muon optimizer as well. These results imply that both gradient orthogonalization and spectral filtering gradients can constrain the activtion-space evolution during training.

6 Conclusion

Topological complexity in activation space is not directly tied to generalization capabilities, as excessive variability undermines performance increases. Increasing activation landscape distance during training does not linearly correlate to improved model accuracy. Diverse exploration of parameter configurations from the Muon optimizer increases validation performance, and may be generally beneficial for modeling multi-modal posteriors.

Limitations and Future Work

Our study is limited by its focus on a single dataset (MNIST) and a simple 2-layer MLP architecture, and the two-point (epochs 0 and 30) persistence snapshots. Additionally, we did not explore the MultiSWAG posterior with topological data analysis tools, only using modes of the posterior to restrict computational complexity.

Future works should explore more experiment setups to ensure results are not an artifact of the dataset and model architectures. The links between in-distribution and out-of-distribution topological invariants and topological complexity and generalization performance should be explored through interventional studies.

Acknowledgements

We thank Keller Jordan for providing the reference implementation of the Muon optimizer and the LBAI Lab team for open-sourcing the Push MultiSWAG code. We acknowledge the developers of the GUDHI, Ripser, and Persim libraries for making topological data analysis tools freely available and well-documented. We thank Tamal Dey and Yusu Wang for their comprehensive textbook (Dey and Wang, 2022), which provided the theoretical backing for our TDA techniques. Finally, we thank Yusu Wang for the course DSC 214: Topological Data Analysis, which made this project possible.

References

Mikhail Belkin. 2021. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. ArXiv:2105.14368 [stat].

Gregory W. Benton, Wesley J. Maddox, Sanae Lotfi, and Andrew Gordon Wilson. 2021. Loss Surface Simplexes for Mode Connecting Volumes and Fast Ensembling. ArXiv:2102.13042 [cs].

Jeremy Bernstein and Laker Newhouse. 2024. Old Optimizer, New Norm: An Anthology. ArXiv:2409.20325 [cs].

Sam Buchanan, Dar Gilboa, and John Wright. 2021. Deep Networks and the Multiple Manifold Problem. ArXiv:2008.11245 [stat].

Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. 2021. Spectral Bias and Task-Model Alignment Explain Generalization in Kernel Regression and Infinitely Wide Neural Networks. *Nature Communications*, 12(1):2914. ArXiv:2006.13198 [stat].

Ting Chen and Tristan Luca Saidi. 2025. Examining the geometry of neural mode connecting loss subspaces.

Tamal K. Dey and Yusu Wang. 2022. *Computational topology for Data Analysis*. Cambridge University Press.

Nikita Doikov, Sebastian U. Stich, and Martin Jaggi. 2024. Spectral Preconditioning for Gradient Methods on Graded Non-convex Functions. ArXiv:2402.04843.

Yangqi Feng, Shing-Ho J. Lin, Baoyuan Gao, and Xian Wei. 2025. Lipschitz Constant Meets Condition Number: Learning Robust and Compact Deep Neural Networks. ArXiv:2503.20454 [cs].

Sara Fridovich-Keil, Raphael Gontijo-Lopes, and Rebecca Roelofs. 2022. Spectral Bias in Practice: The Role of Function Frequency in Generalization. ArXiv:2110.02424.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. 2019. An Investigation into Neural Net Optimization via Hessian Eigenvalue Density.

giotto-tda contributors. 2023. Betticurve - giotto-tda documentation. `https://giotto-ai.github.io/gtda-docs/latest/modules/generated/diagrams/representations/gtda.diagrams.BettiCurve.html`.

Vineet Gupta, Tomer Koren, and Yoram Singer. 2018. Shampoo: Preconditioned Stochastic Tensor Optimization. ArXiv:1802.09568 [cs].

Lei Huang, Li Liu, Fan Zhu, Diwen Wan, Zehuan Yuan, Bo Li, and Ling Shao. 2020. Controllable Orthogonalization in Training DNNs. ArXiv:2004.00917 [cs].

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. The Platonic Representation Hypothesis. ArXiv:2405.07987 [cs].

Keller Jordan. 2024. Muon: An optimizer for hidden layers in neural networks | Keller Jordan blog.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. 2025. Muon is Scalable for LLM Training. ArXiv:2502.16982 [cs].

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Wesley Maddox, Timur Garipov, Pavel Izmailov, Dmitry Vetrov, and Andrew Gordon Wilson. 2019. A simple baseline for bayesian uncertainty in deep learning.

Jialin Mao, Itay Griniasty, Han Kheng Teoh, Rahul Ramesh, Rubing Yang, Mark K. Transtrum, James P. Sethna, and Pratik Chaudhari. 2024. The Training Process of Many Deep Networks Explores the Same Low-Dimensional Manifold. *Proceedings of the National Academy of Sciences*, 121(12):e2310002121. ArXiv:2305.01604 [cs].

Audun Myers, Elizabeth Munch, and Firas A. Khasawneh. 2019. Persistent homology of complex networks for dynamic state detection. *Phys. Rev. E*, 100:022314.

Samet Oymak and Mahdi Soltanolkotabi. 2018. Overparameterized Nonlinear Learning: Gradient Descent Takes the Shortest Path? ArXiv:1812.10004 [cs].

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. 2019. On the Spectral Bias of Neural Networks. ArXiv:1806.08734 [stat].

Gurjeet Singh, Facundo Mémoli, and Gunnar Carlsson. 2007. Topological methods for the analysis of high dimensional data sets and 3d object recognition. pages 91–100.

Jarkko Venna and Samuel Kaski. 2001. Neighborhood preservation in nonlinear projection methods: An experimental study. In *Artificial Neural Networks — ICANN 2001*, pages 485–491, Berlin, Heidelberg. Springer Berlin Heidelberg.

Andrew Gordon Wilson and Pavel Izmailov. 2022. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. ArXiv:2002.08791 [cs].

Mitchell Wortsman, Maxwell Horton, Carlos Guestrin, Ali Farhadi, and Mohammad Rastegari. 2021. Learning Neural Network Subspaces. ArXiv:2102.10472 [cs].

Greg Yang, James B. Simon, and Jeremy Bernstein. 2024. A Spectral Condition for Feature Learning. ArXiv:2310.17813 [cs].

David Yunis, Kumar Kshitij Patel, Samuel Wheeler, Pedro Savarese, Gal Vardi, Karen Livescu, Michael Maire, and Matthew R. Walter. 2024. Approaching Deep Learning through the Spectral Dynamics of Weights. ArXiv:2408.11804 [cs].

Berfin Şimşek, François Ged, Arthur Jacot, Francesco Spadaro, Clément Hongler, Wulfram Gerstner, and Johanni Brea. 2021. Geometry of the Loss Landscape in Overparameterized Neural Networks: Symmetries and Invariances. ArXiv:2105.12221 [cs].

A Pretrain Accuracy

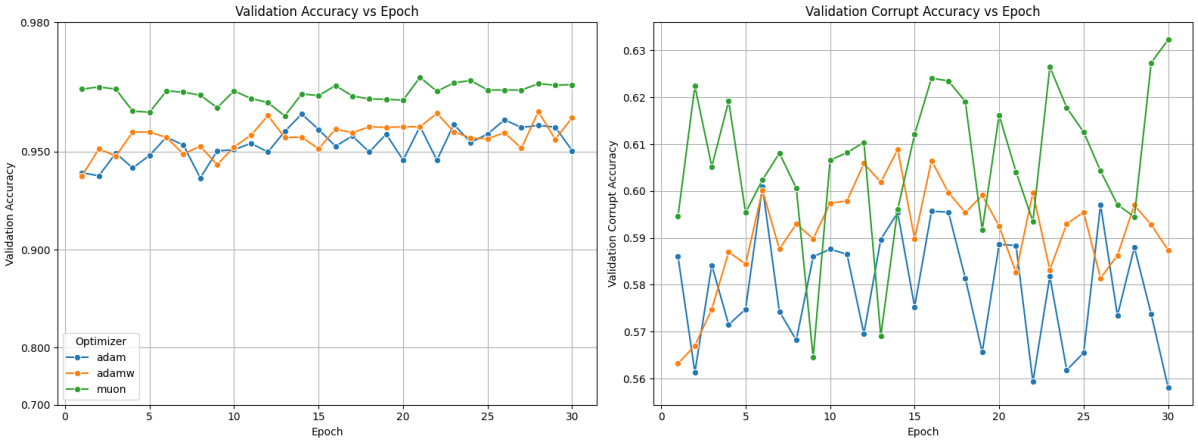

Figure 2: Validation Accuracy for Regular Optimizers

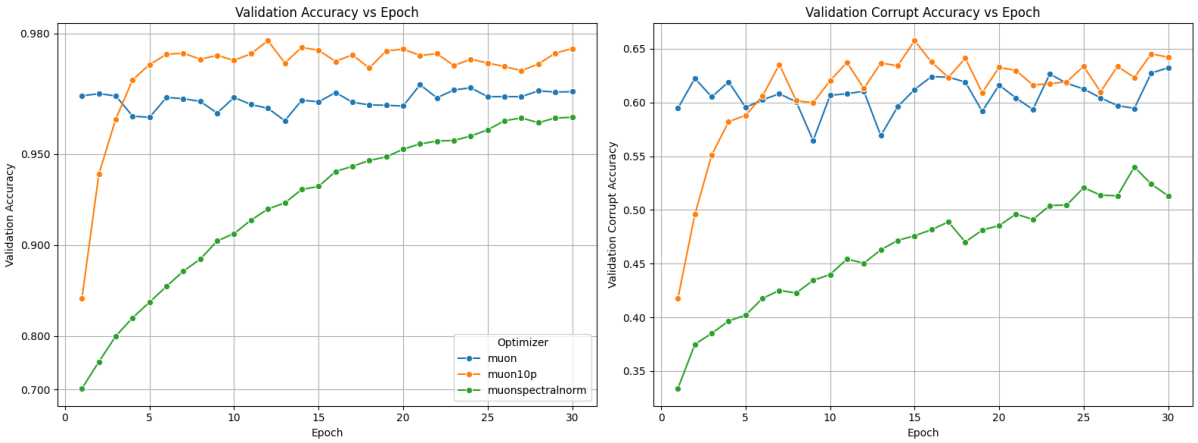

Figure 3: Validation Accuracy for Muon Ablations

B Pretrain Ensemble Aggregate Total L^2 Gradient Norm

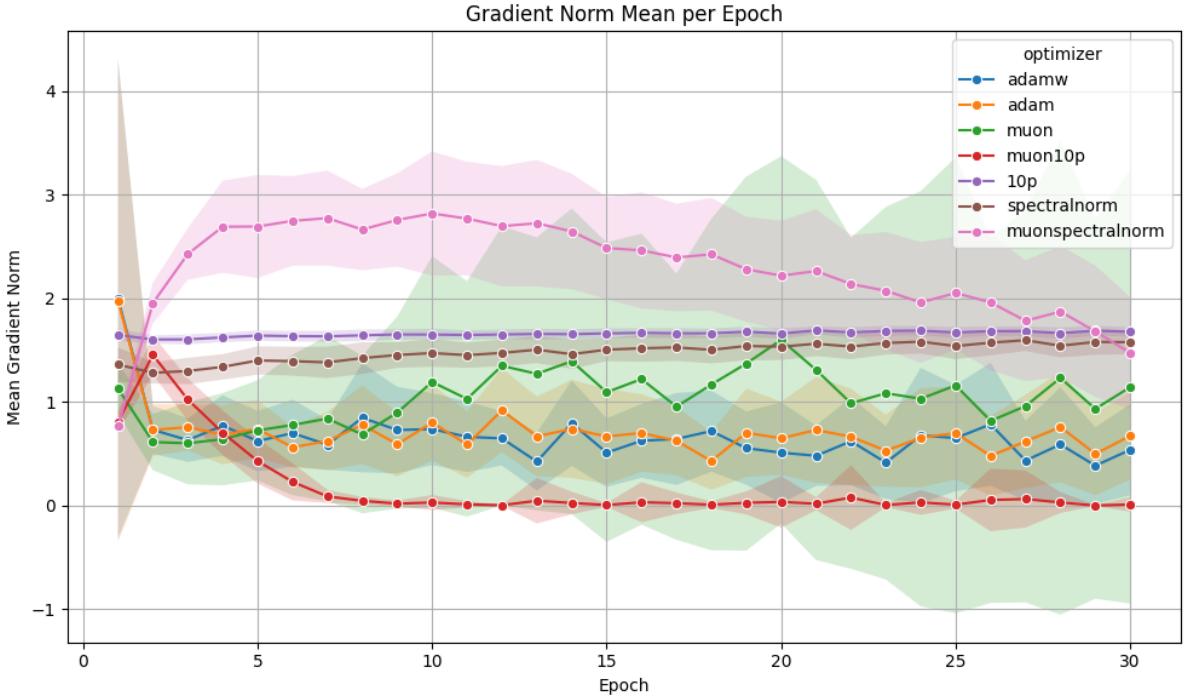

Figure 4: Gradient Mean of Total L^2 Norm per Epoch

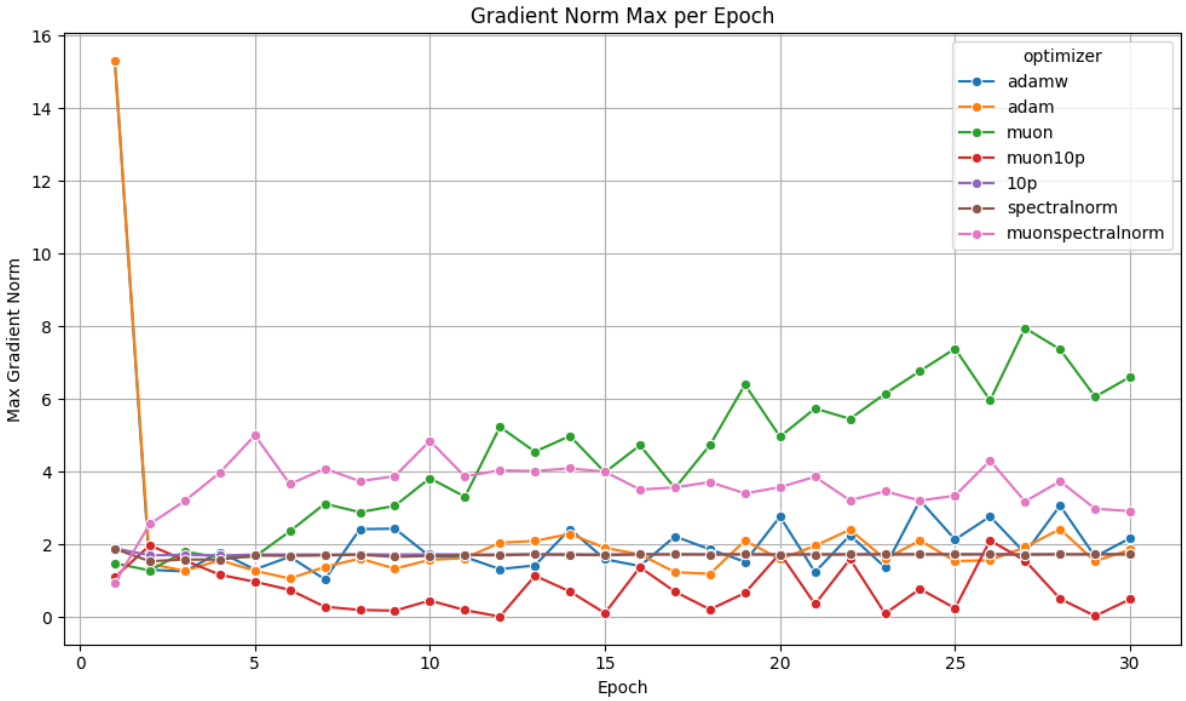

Figure 5: Gradient Max of Total L^2 Norm per Epoch

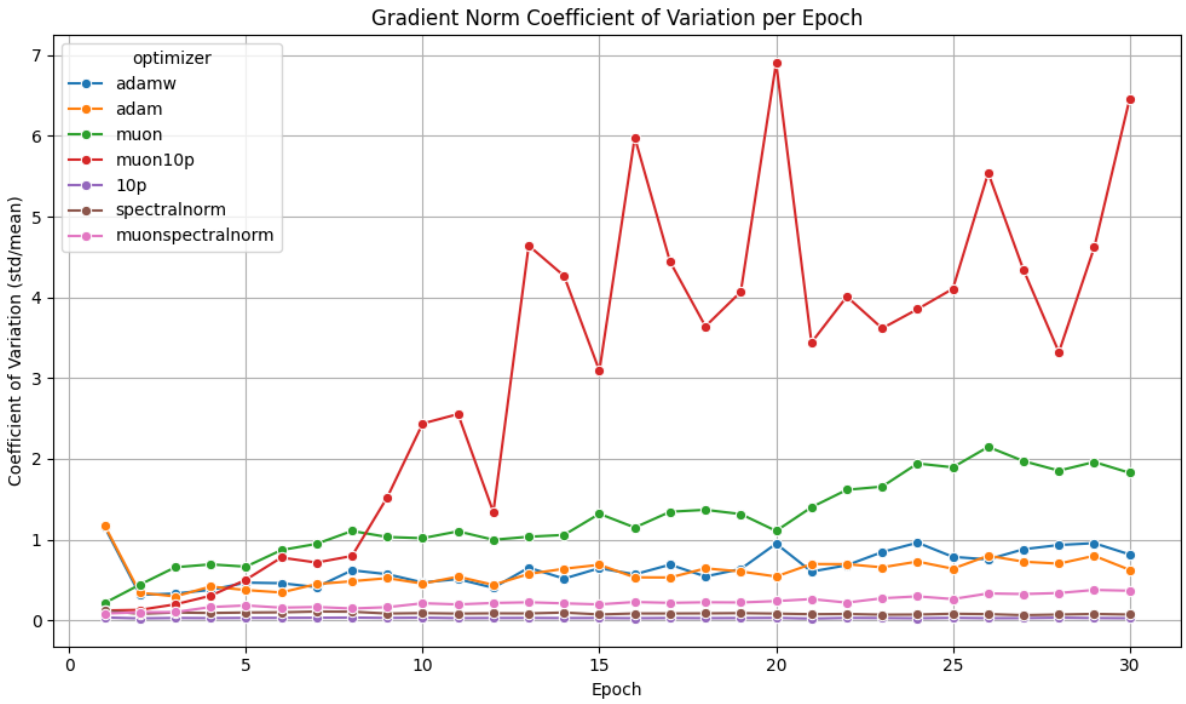

Figure 6: Grad Coefficient of Variation Total L^2 Norm per Epoch

C Pretrain Training Losses

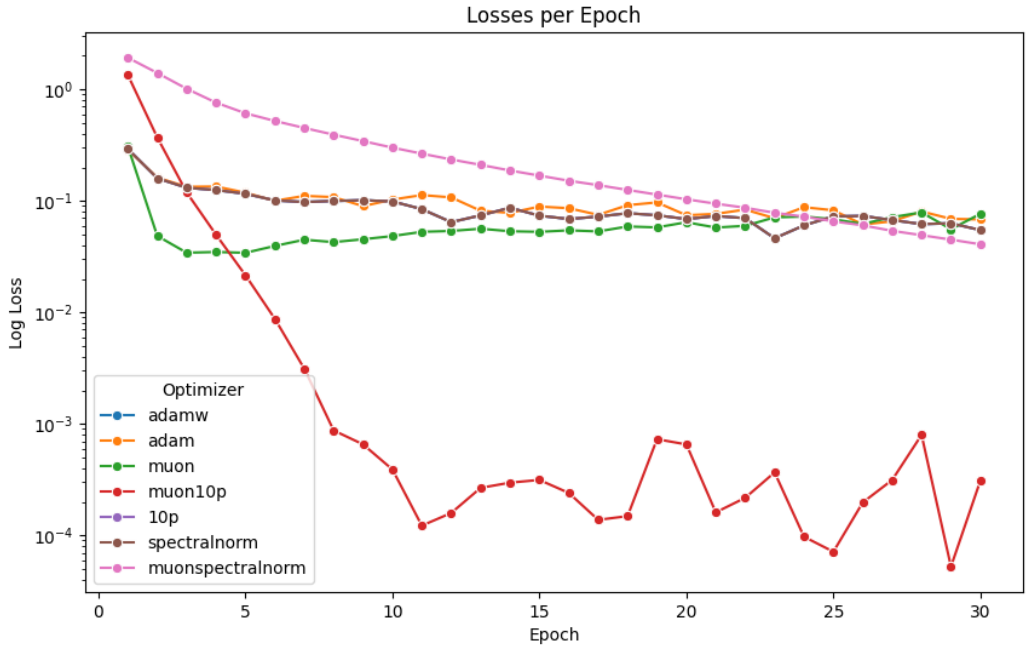

Figure 7: Pretrain Losses on Training Dataset

D SWAG Covariance Matrix Statistics

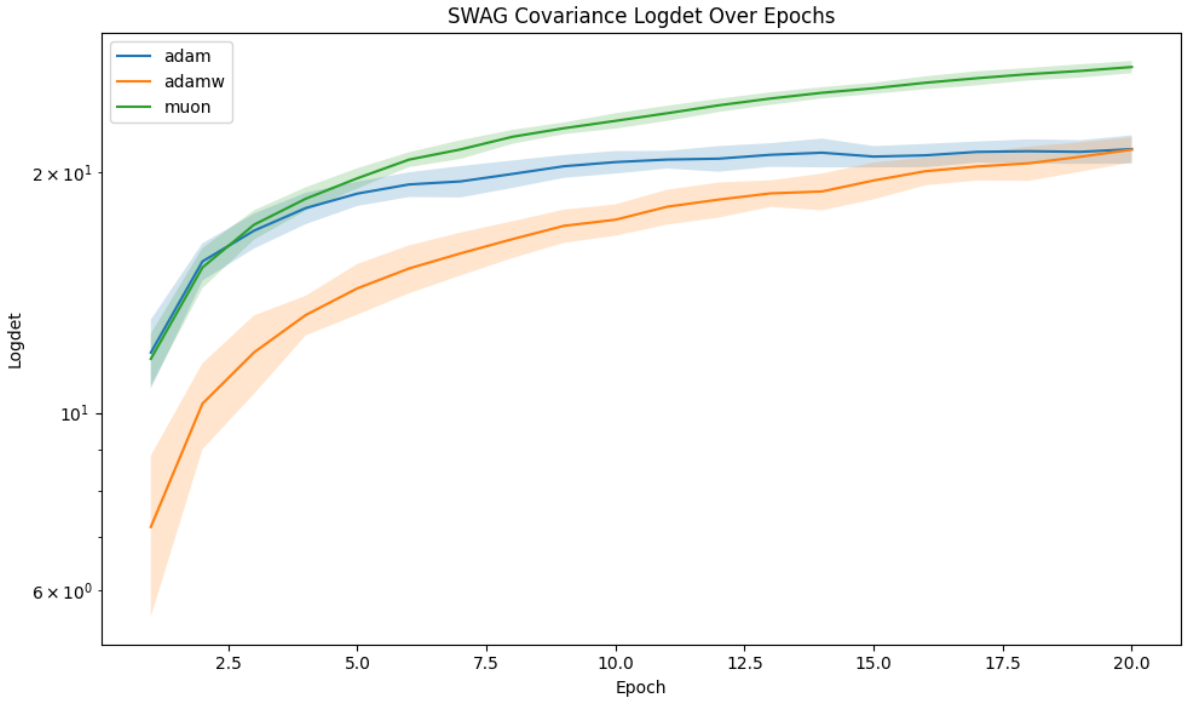

Figure 8: SWAG Covariance Matrix Log-Determinant

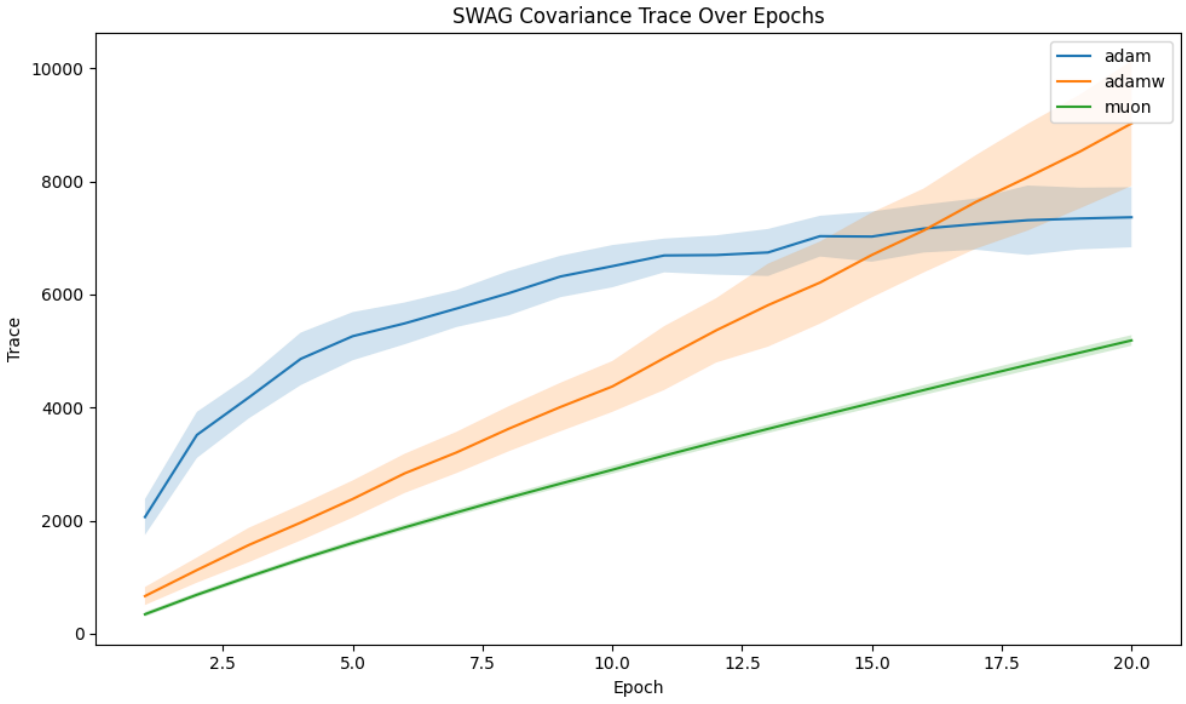

Figure 9: SWAG Covariance Matrix Trace

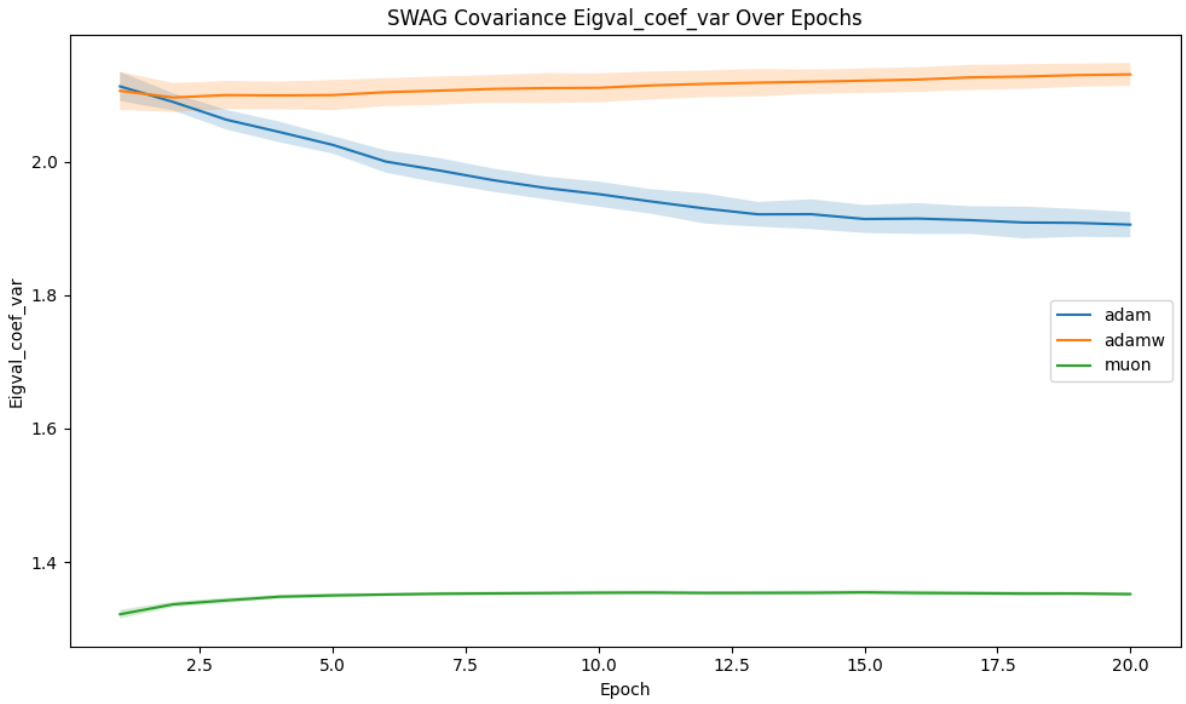

Figure 10: SWAG Coefficient of Variation of Covariance Matrix Eigenvalues

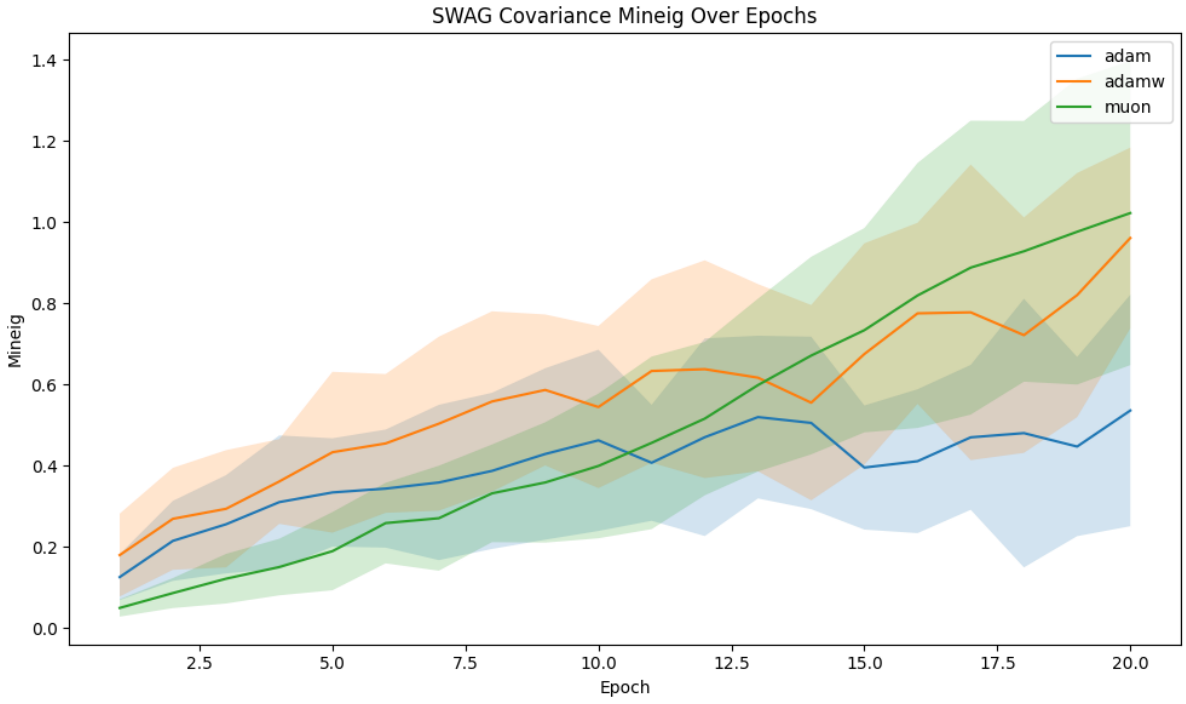

Figure 11: SWAG Minimum of Covariance Matrix Eigenvalues

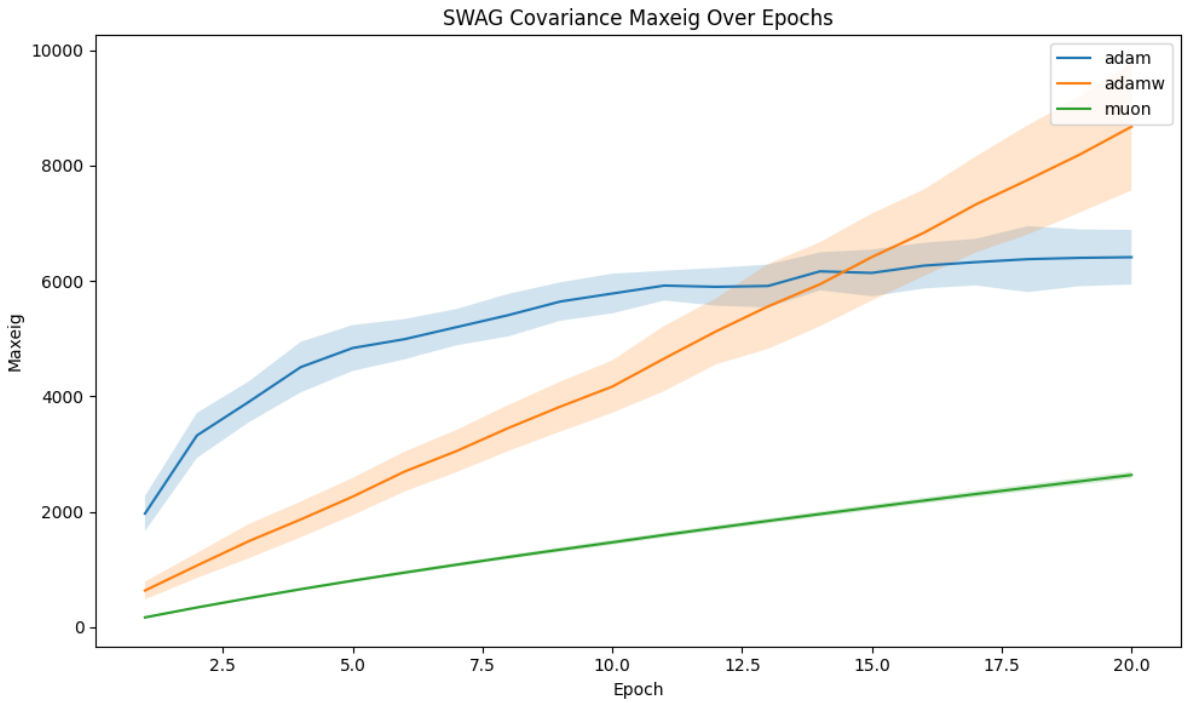

Figure 12: SWAG Maximum of Covariance Matrix Eigenvalues

E Mapper Graph of Parameter Trajectories

For each of these Mapper graphs, we use the UMAP lens on a Point Cloud made the top 50 principal components of the flattened parameters for each particle for each epoch of pretraining. We use 5 cubes with 50% overlap for our cover.

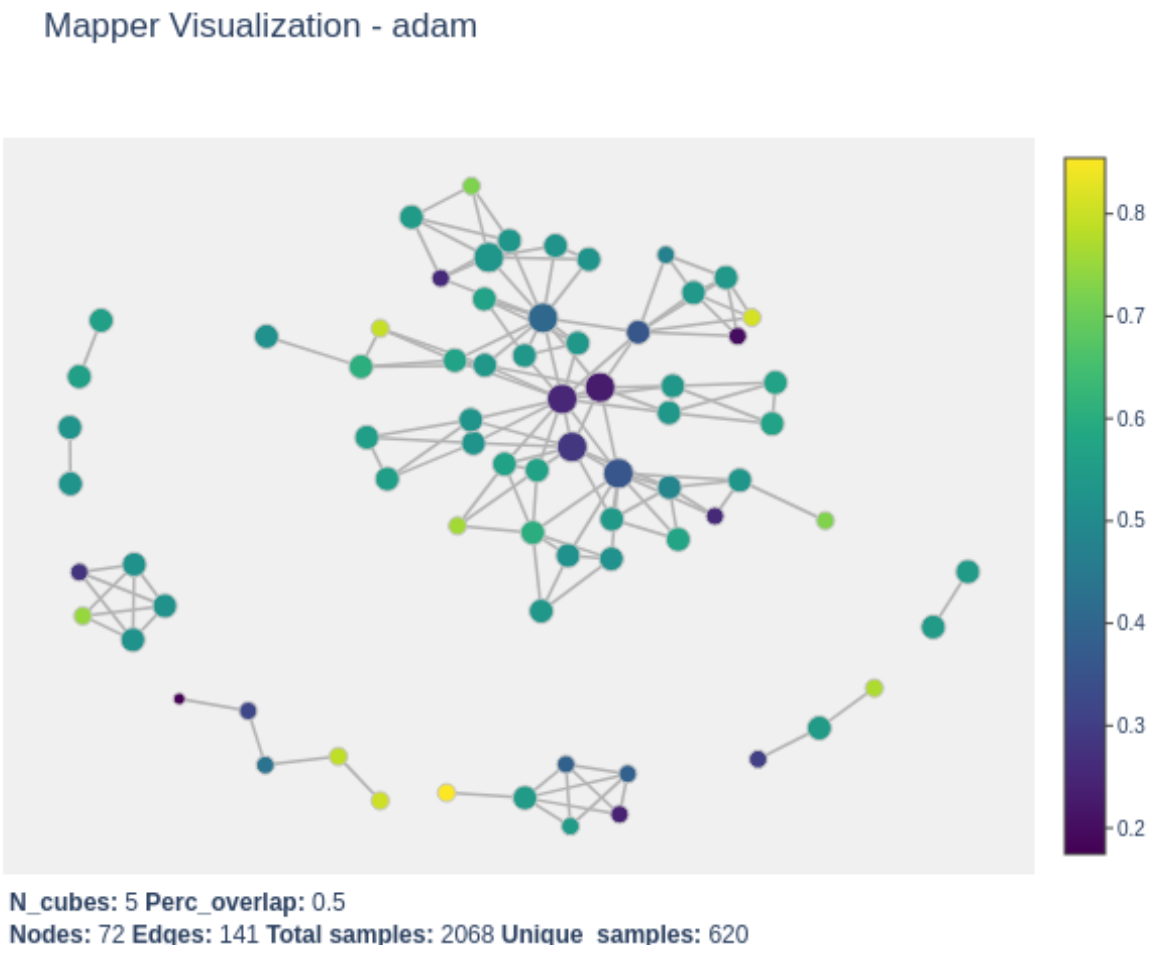

Figure 13: Mapper Graph for Adam Trajectories

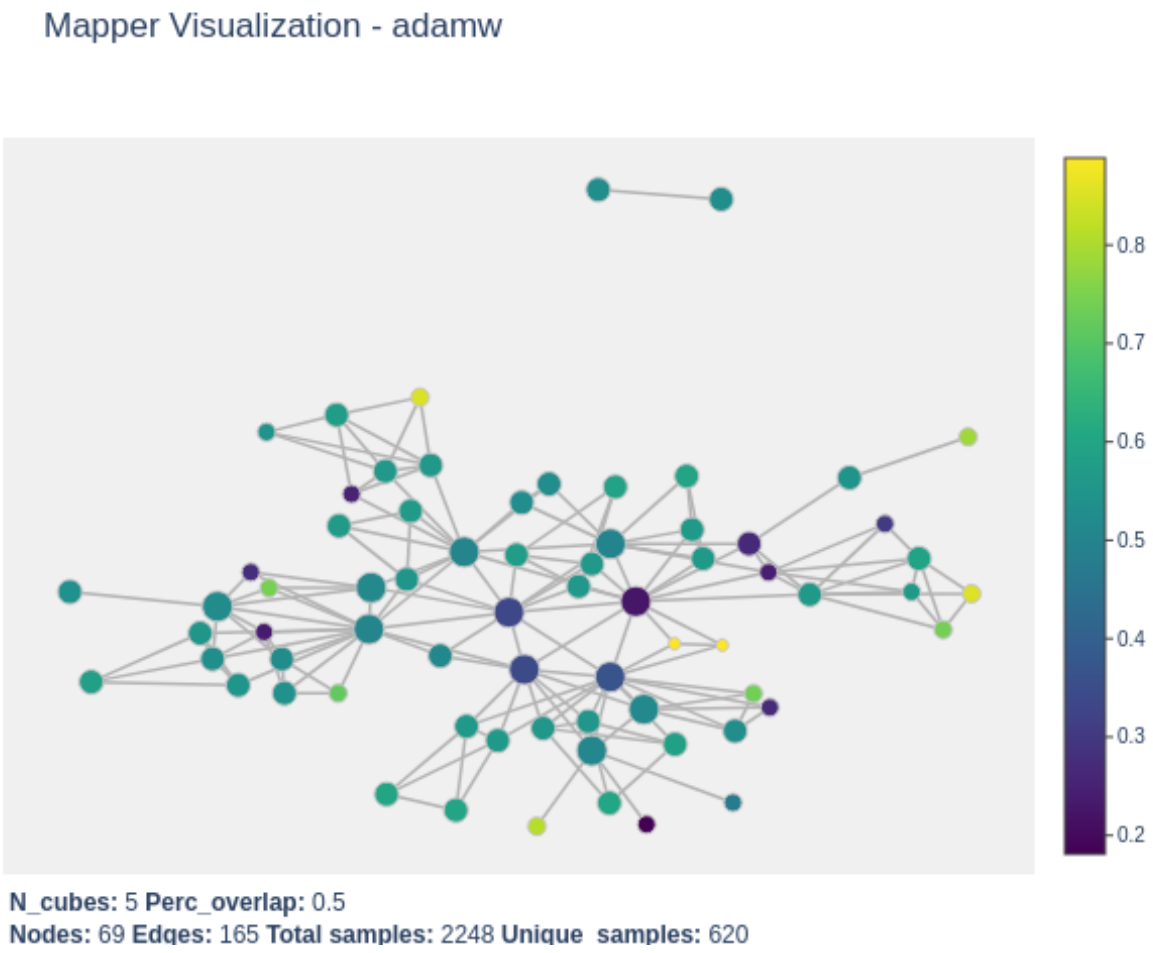

Figure 14: Mapper Graph for AdamW Trajectories

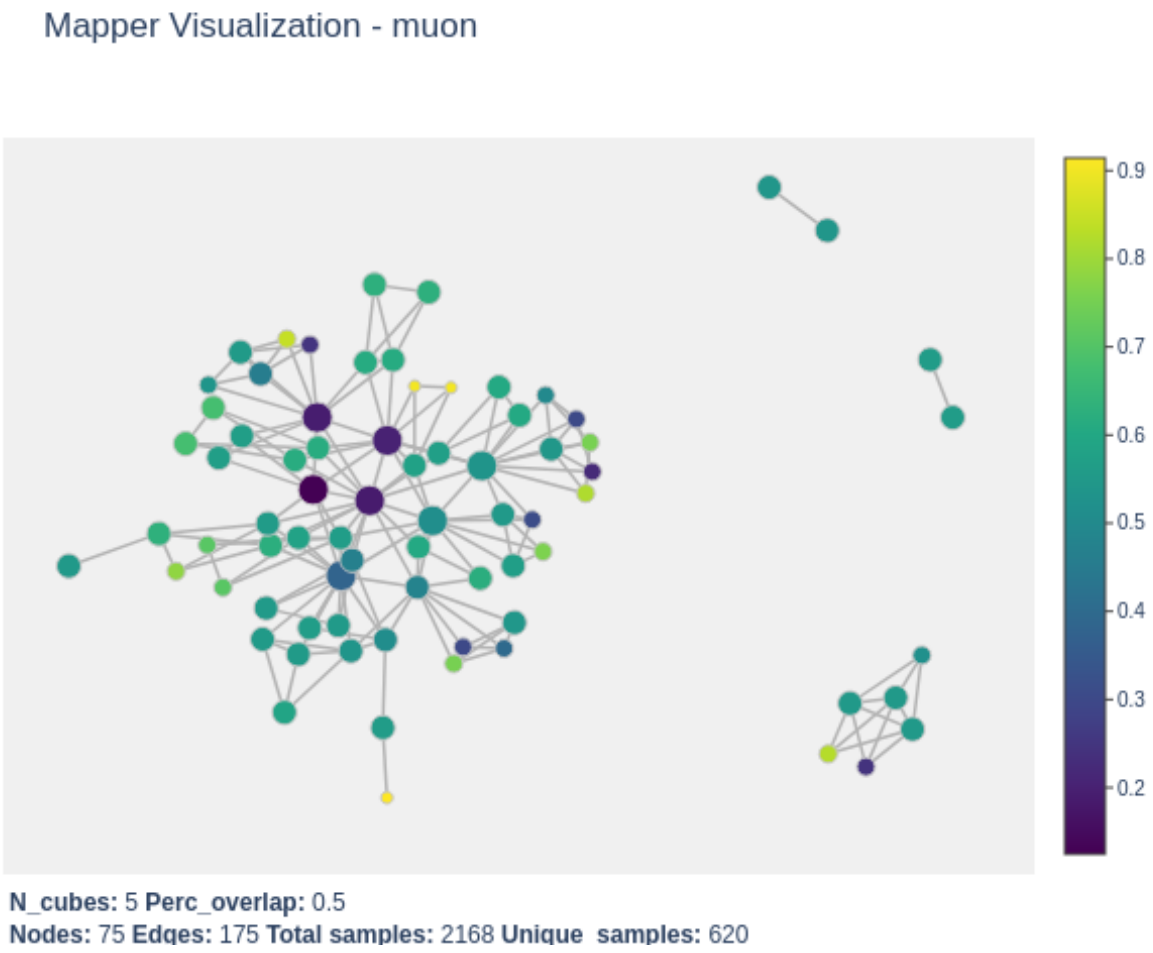

Figure 15: Mapper Graph for Muon Trajectories

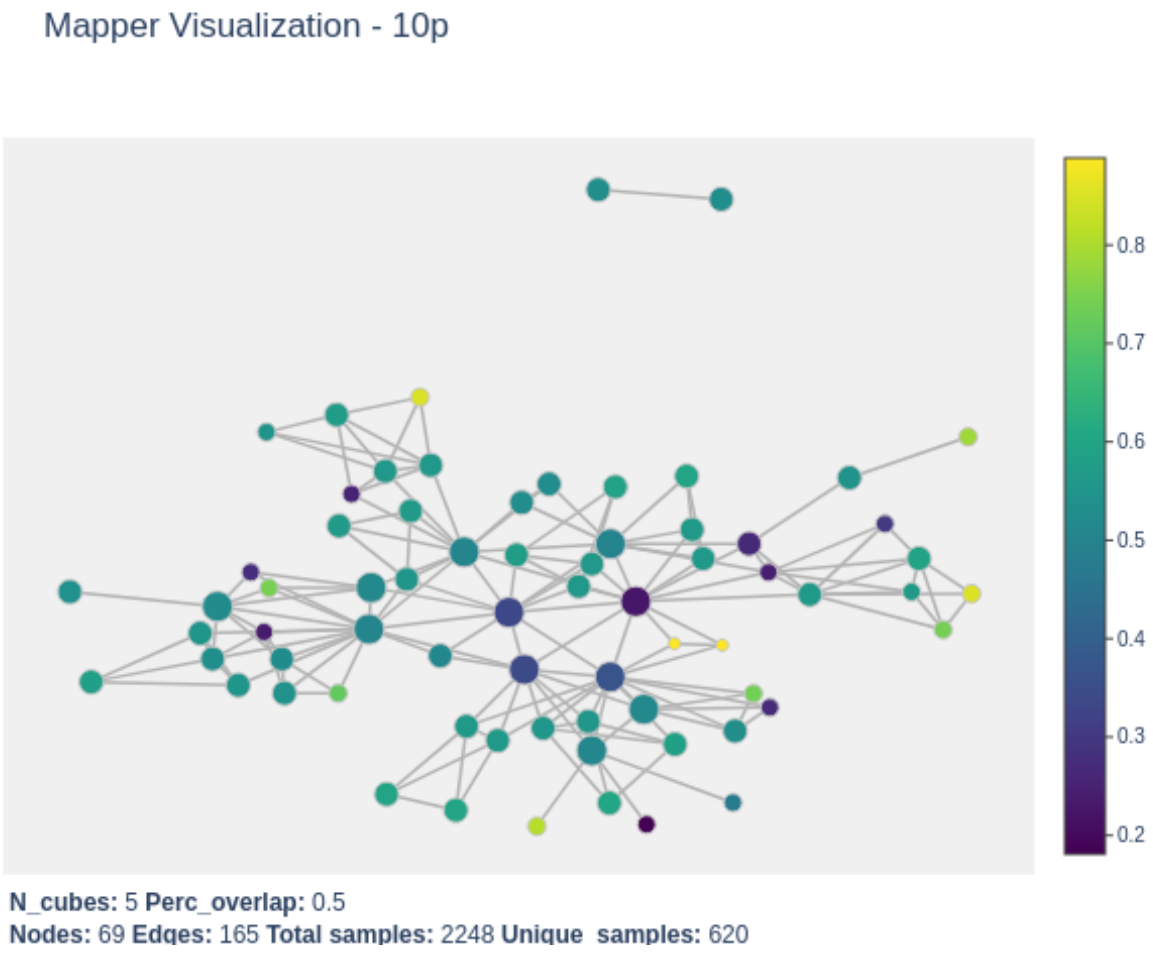

Figure 16: Mapper Graph for 10p Trajectories

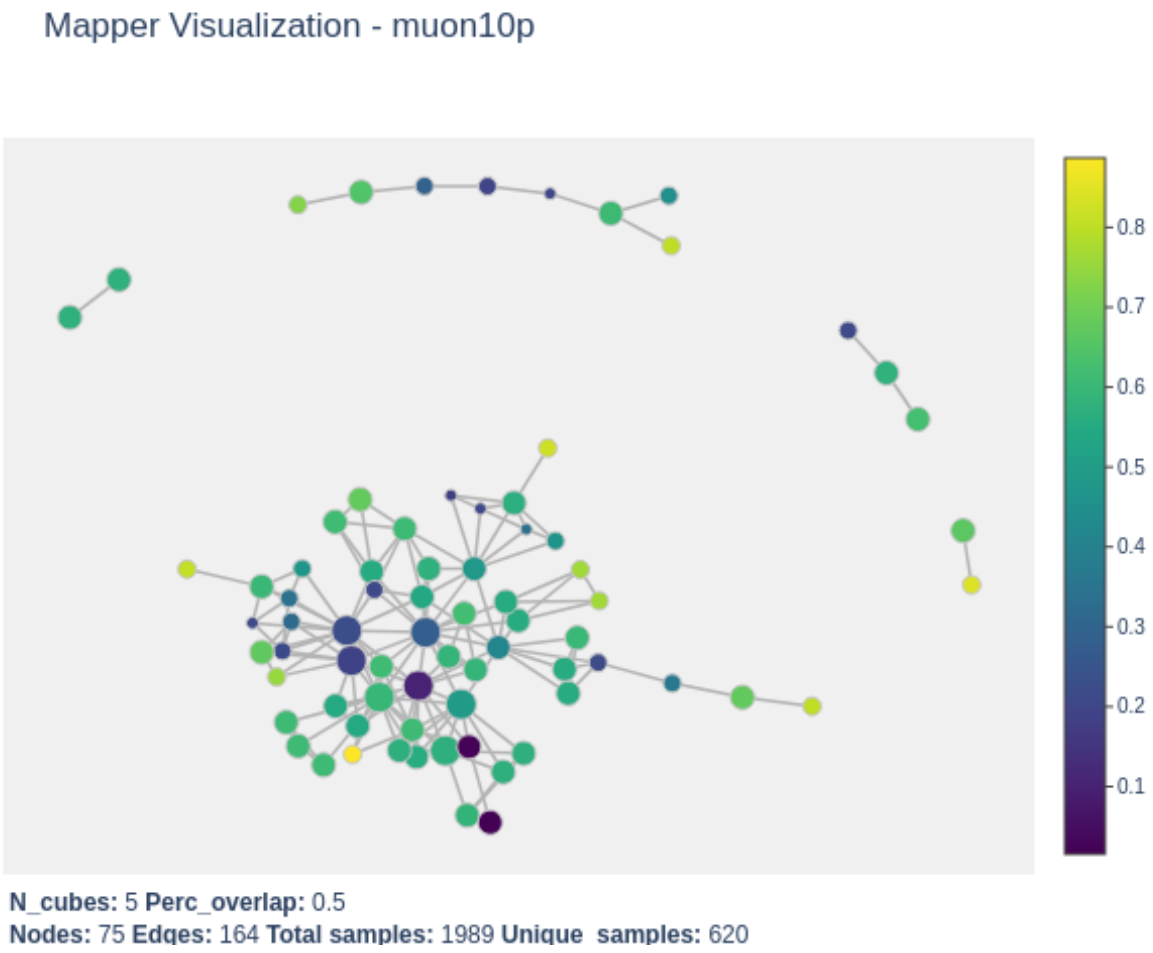

Figure 17: Mapper Graph for Muon10p Trajectories

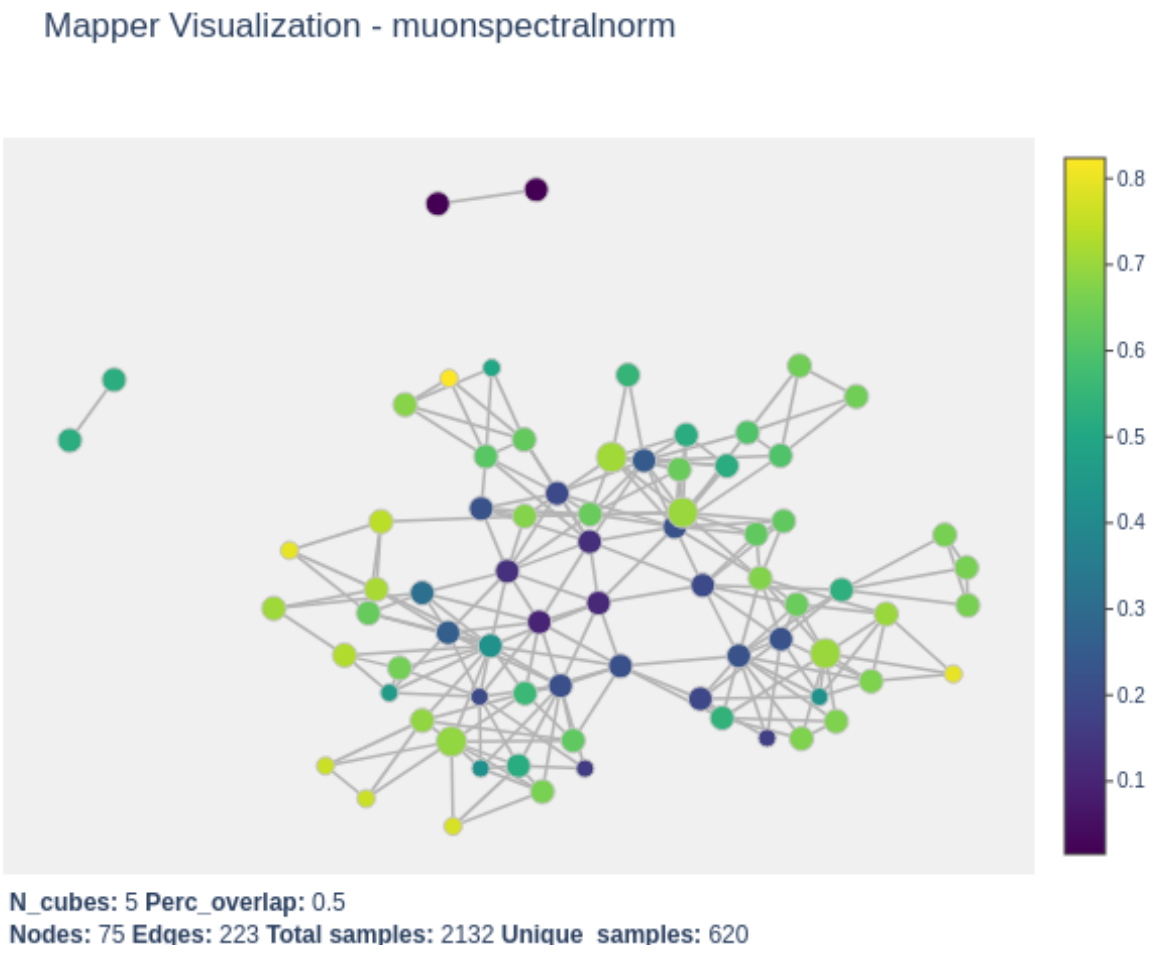

Figure 18: Mapper Graph for Muon Spectral Norm Trajectories

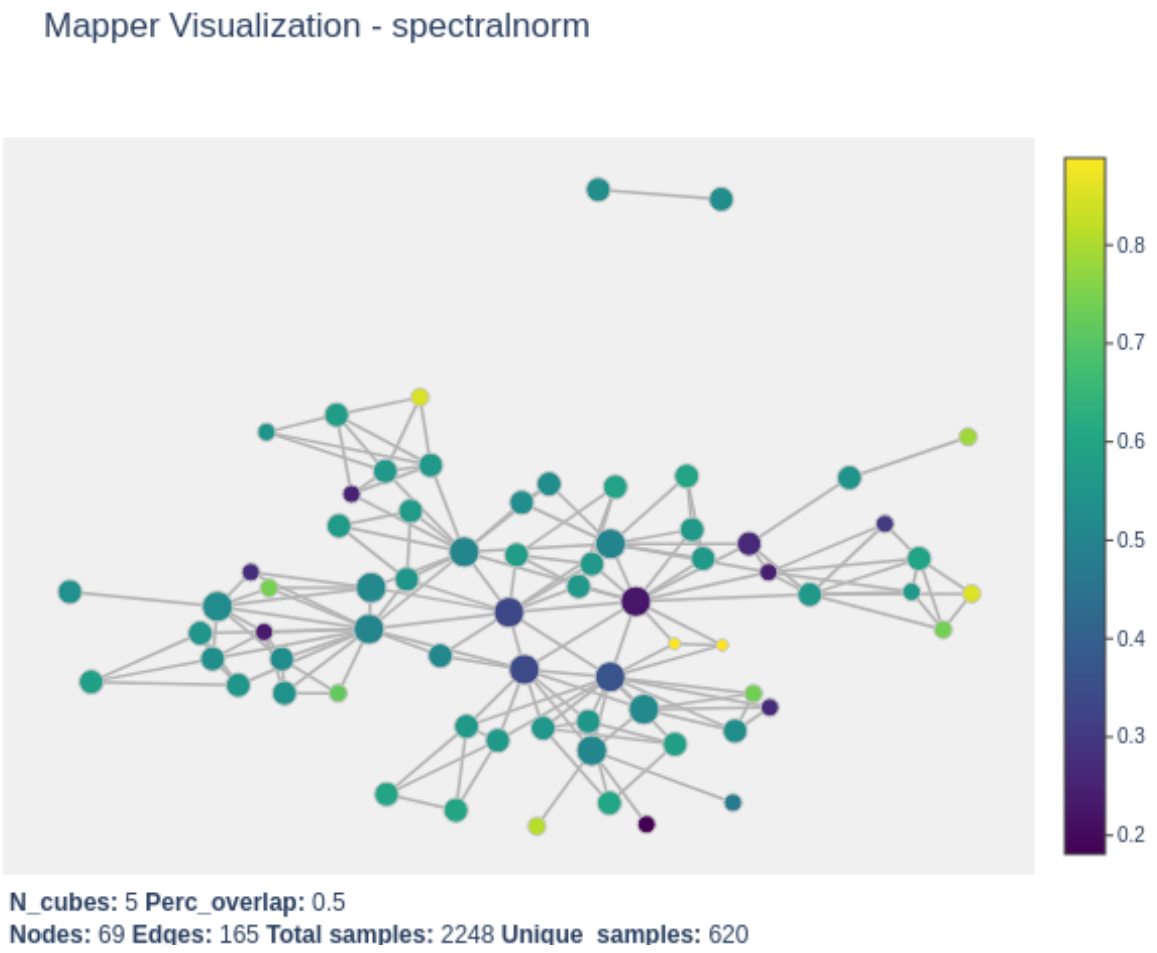

Figure 19: Mapper Graph for Spectral Norm Trajectories

F Penultimate Activation Landscapes

F.1 Landscape Stability

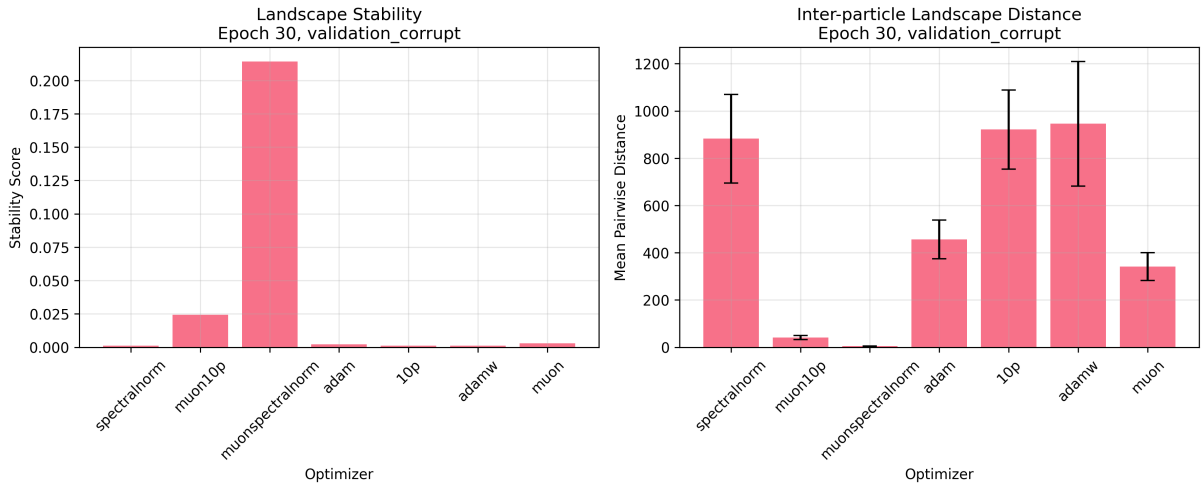

Figure 20: Corrupt Stability

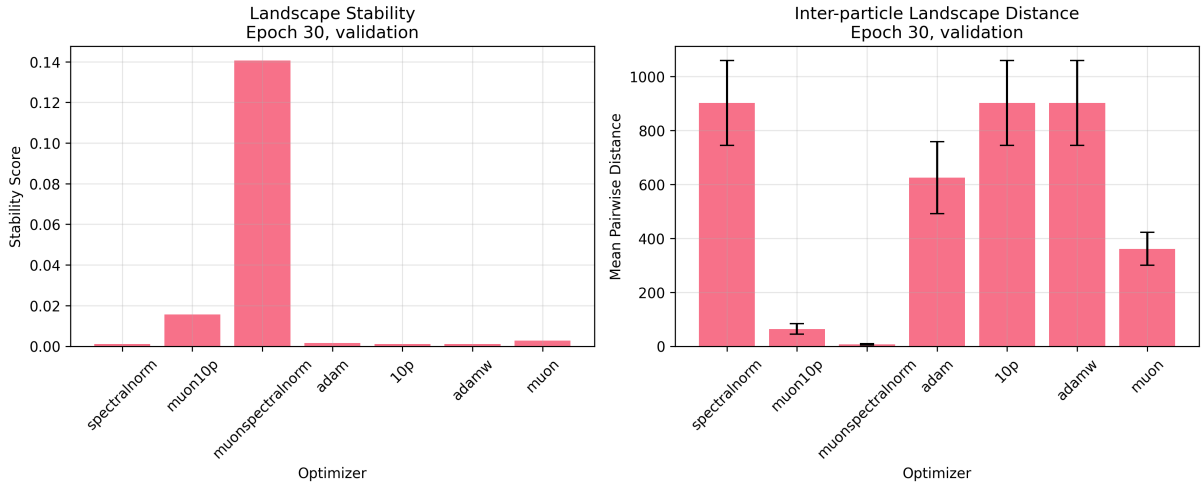

Figure 21: Stability

F.2 Landscape Time Series

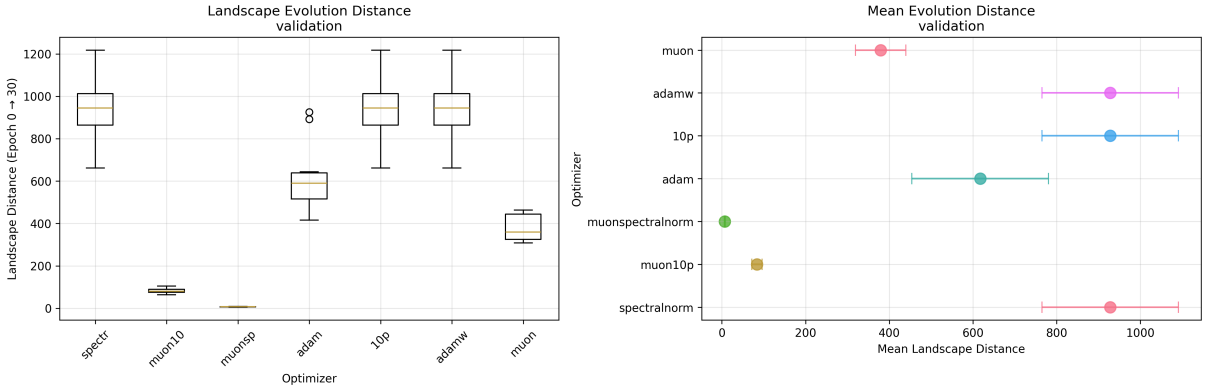

Figure 22: Time Series

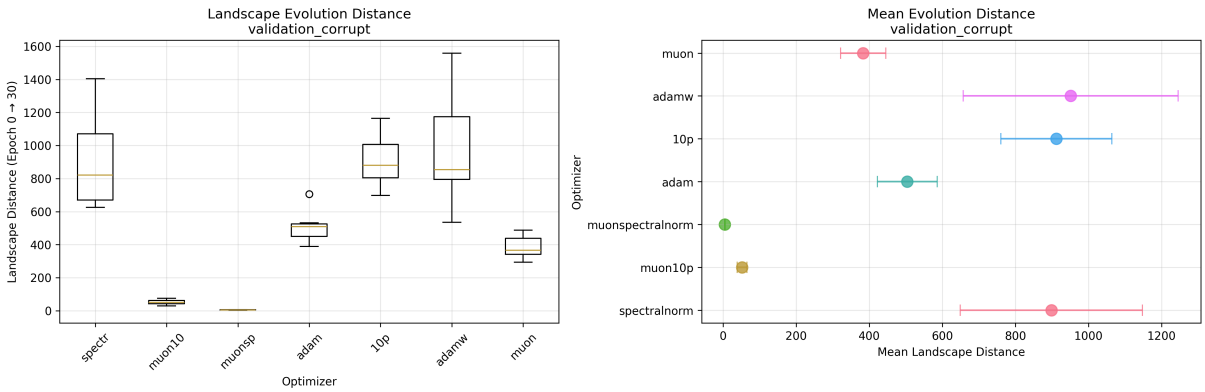

Figure 23: Time Series Corrupt

F.3 Landscape Distances

Figure 24: Landscape Distances

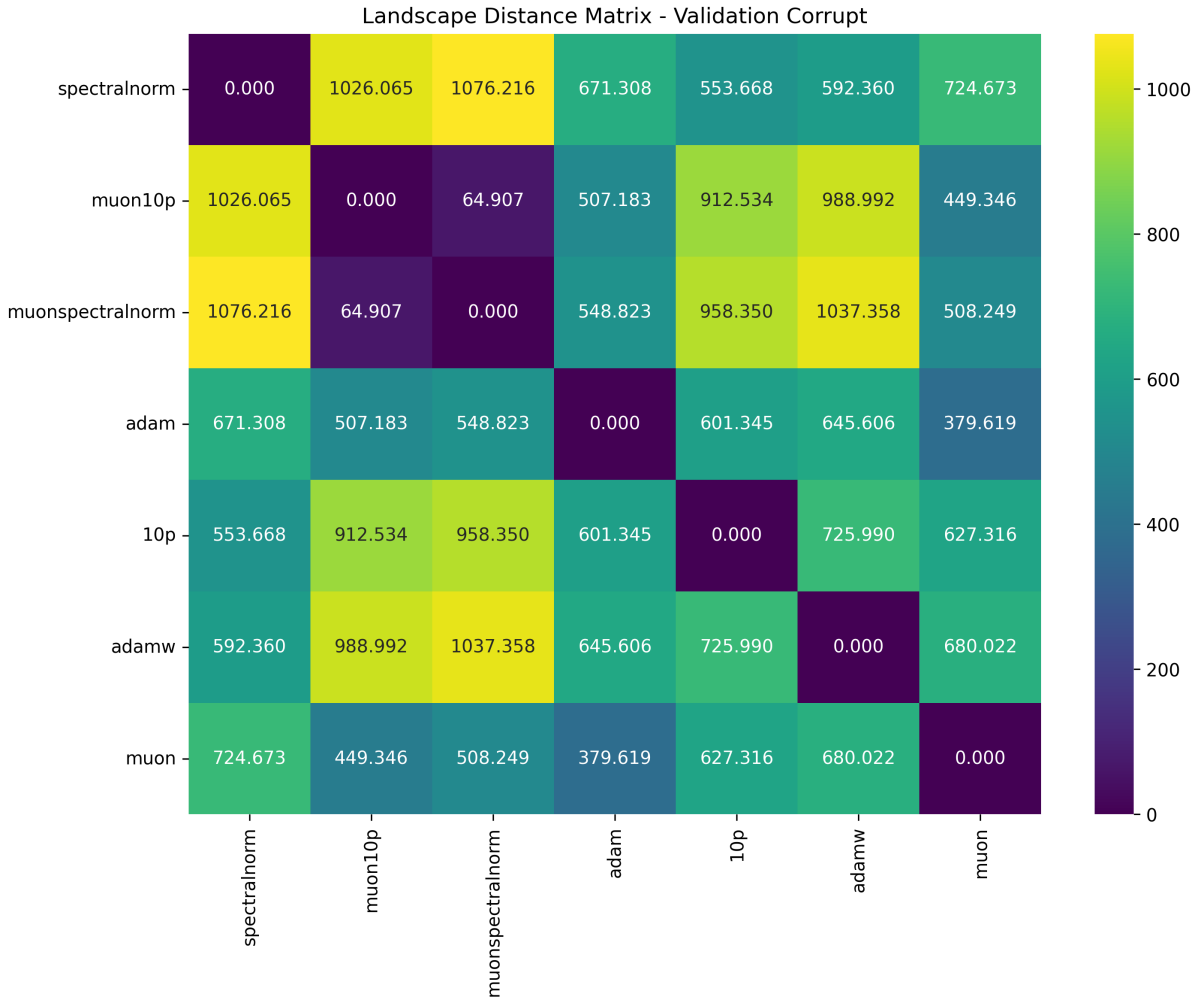

Figure 25: Landscape Distances Corrupt

F.4 Betti Curves

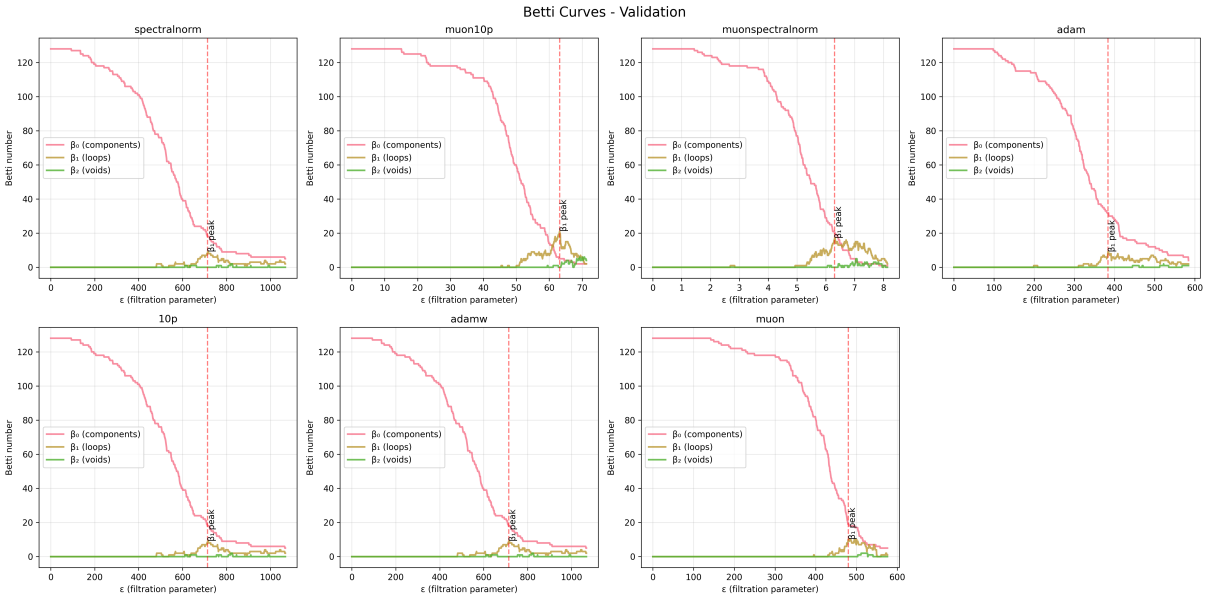

Figure 26: Betti Curves

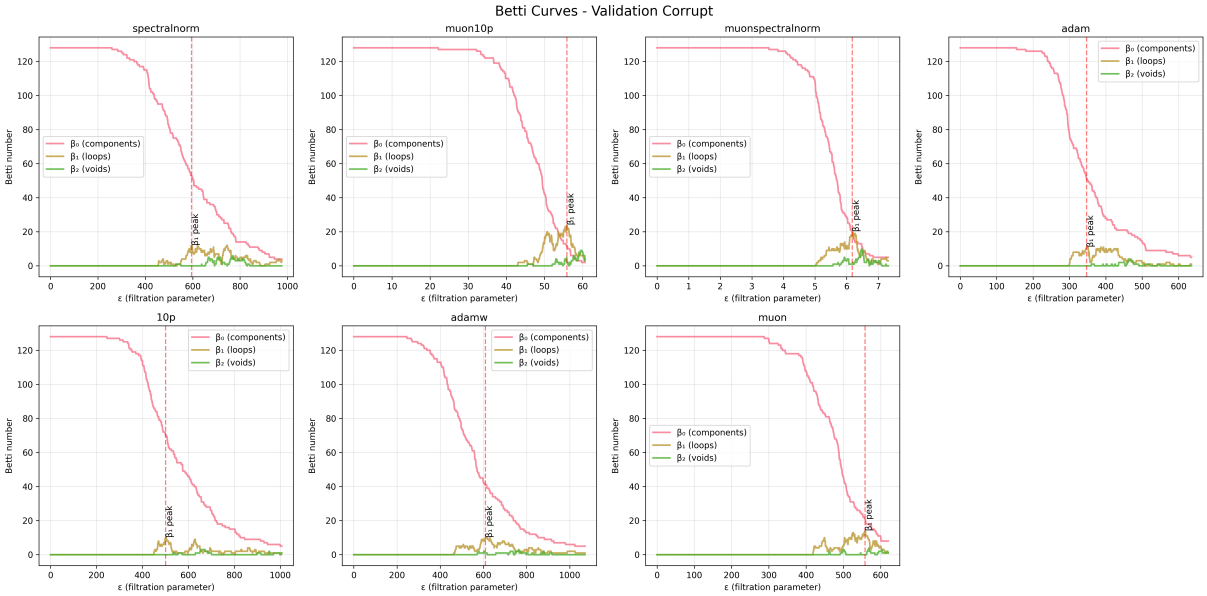

Figure 27: Betti Curves Corrupt