Orthogonal Re-basin ID4

November 2, 2025

Julian Hendrix

Abstract

This project investigates whether orthogonal transformations can be used to align independently trained neural networks for smooth weight-space interpolation. We focus on ResNet-20 models trained on CIFAR-10 and compare Orthogonal Re-Basin, a relaxation of Git Re-Basin that allows full orthogonal transformations against a naive model interpolation baseline and Git Re-Basin. Contrary to expectations, orthogonal alignment does not reduce interpolation loss barriers nor improve representation similarity. We report diagnostic metrics showing that both alignment strategies degrade functional similarity compared to the unaligned baseline.

1. Introduction

Independently trained models often converge to different points in weight space, but may still encode similar functions. Aligning them could enable weight-space model ensembling, federated learning without sharing data, and training-efficient model soups. Prior work has shown that low-loss linear interpolation is possible after permutation-based neuron-matching. In this project, we investigate a more general alignment approach using orthogonal matrices instead of permutations, aiming to enable smoother merging trajectories than Git Re-Basin.

2. Related Work

The *Git Re-Basin* framework (Ainsworth et al., 2023) aligns networks by computing layer-wise neuron permutations that maximize representation overlap. Recent geometric approaches (Singh et al., 2024; **?**) suggest using orthogonal transformations instead. Other works have explored mode connectivity (Garipov et al., 2018) and interpolation between model parameters (Wortsman et al., 2022), while *CKA* (Kornblith et al., 2019) is widely used to quantify functional similarity between neural representations.

3. Method

We train two independent ResNet-20 models on CIFAR-10 (test accuracy $\approx 88\%$). To align models A and B at a given layer, we compute an orthogonal matrix R^* solving the Procrustes problem:

$$R^* = \arg\min_{R \in O(d)} \|XR - Y\|_F^2 \tag{1}$$

where X and Y are activation matrices from the same layer on shared data and $O(d)$ is the orthogonal group. We apply R^* to the incoming weights of layer B and its transpose to the outgoing weights. For comparison, Git Re-Basin computes permutation matrices via matching.

We evaluate weight-space interpolation:

$$\theta_\alpha = (1-\alpha)\theta_A + \alpha\theta_B$$

using both linear (LERP) and spherical (SLERP) interpolation. Metrics include:

- **Loss barrier**: $\max_\alpha \mathcal{L}(\theta_\alpha)$
- **Midpoint accuracy**: accuracy at $\alpha = 0.5$
- **CKA similarity**: functional similarity of embeddings
- **Cycle-consistency error**: deviation from identity after round-trip mapping
- **Residual misalignment error (RME)**: due to non-commutativity with ReLU

4. Experimental Setup

Both ResNet-20 models were trained from scratch using SGD with momentum (0.9), weight decay ($5\text{e}{-4}$), and cosine learning rate decay. Models were trained for 200 epochs on CIFAR-10 with batch size 128 and random data augmentations. We use 5,000 shared samples to compute transformations.

Email: Julian Hendrix <hendrix.2090880@studenti.uniroma1.it>.

Machine Learning 2025, Sapienza University of Rome, 2nd semester a.y. 2024/2025.

5. Results

5.1. Interpolation Performance

Table 1 shows loss barrier and midpoint accuracy. Surprisingly, naive interpolation performs best. Orthogonal Re-Basin drastically increases the loss barrier and collapses midpoint accuracy.

Table 1. Interpolation results. Lower loss barrier is better; higher midpoint accuracy is better.

Method	Loss Barrier ↓	Midpoint Acc. ↑
Naive + Linear	**2.18**	10.9%
Naive + SLERP	3.55	**14.3**%
Git Re-Basin + Linear	6.20	10.0%
Orthogonal Re-Basin + Linear	159.80	10.0%
Orthogonal Re-Basin + SLERP	159.80	10.0%

5.2. Diagnostics

Table 2 reports CKA similarity and alignment errors. Orthogonal Re-Basin reduces similarity and produces a large cycle-consistency error and non-zero RME, suggesting representational distortion.

Table 2. Diagnostic metrics. Higher CKA, lower errors are better.

Method	CKA ↑	Cycle-Cons. ↓	RME ↓
Naive	**0.727**	N/A	N/A
Git Re-Basin	0.361	N/A	N/A
Orthogonal Re-Basin	0.302	114.05	0.0677

5.3. Plots

We visualize test loss and accuracy along the interpolation paths for all methods.

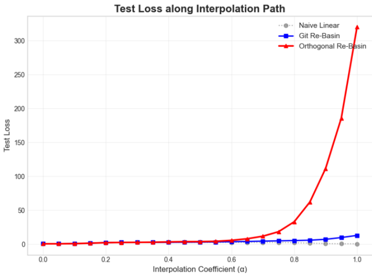

Figure 1. Test loss across interpolation paths. Orthogonal alignment collapses performance.

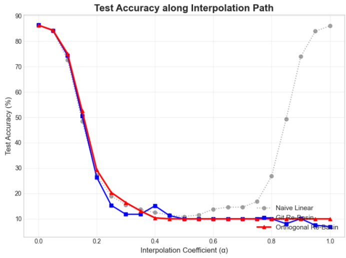

Figure 2. Test accuracy across interpolation paths. Orthogonal alignment collapses performance.

6. Discussion & Future Work

Orthogonal alignment was expected to improve interpolation by capturing more flexible symmetry than permutation. Instead, both Git and Orthogonal Re-Basin degraded similarity and raised loss barriers. Likely causes:

- Incorrect handling of BatchNorm and biases during transformation
- Misalignment of ReLU due to non-commutativity
- Failure to re-normalize activations post-transformation

Future work: (i) incorporate BatchNorm correction, (ii) fine-tune models after alignment, (iii) constrain R^* via CKA regularization to preserve functionality, (iv) test on MLPs or ViTs to isolate architectural effects.

Bibliography. (Ainsworth et al., 2023; Singh et al., 2024; Garipov et al., 2018; Wortsman et al., 2022; Kornblith et al., 2019; Anonymous, 2024b;a; Wen & Yin, 2013; noe, 2021; sym, 2025; sym; opt, 2022; bey, 2023; fer, 2024; dra, 2018a;b; lin; equ; coh, 2018; awe, 2025; eme, 2024; nvi, 2024; sle, 2025)

References

More efficient training using equivariant neural networks. https://uu.diva-portal.org/smash/get/diva2:1779131/FULLTEXT01.pdf. Accessed: October 25, 2025.

Linear mode connectivity between multiple models modulo permutation symmetries. https://openreview.net/forum?id=qaJuLzY6lI. Accessed: October 25, 2025.

Symmetries of neural networks. http://bactra.org/notebooks/symmetries-of-neural-networks.html. Accessed: October 25, 2025.

On the generalization of equivariance and convolution in neural networks. *arXiv preprint arXiv:1802.03690*, 2018. URL https://arxiv.org/pdf/1802.03690. Accessed: October 25, 2025.

Essentially no barriers in neural network energy landscape. *arXiv preprint arXiv:1803.00885*, 2018a. URL https://arxiv.org/pdf/1803.00885. Accessed: October 25, 2025.

Essentially no barriers in neural network energy landscape. In *Proceedings of Machine Learning Research*, 2018b. URL https://proceedings.mlr.press/v80/draxler18a.html. Accessed: October 25, 2025.

Noether's learning dynamics: Role of symmetry breaking in neural networks. In *NeurIPS*, 2021. URL https://proceedings.neurips.cc/paper/2021/file/d76d8deea9c19cc9aaf2237d2bf2f785-Paper.pdf. Accessed: October 25, 2025.

On convexity and linear mode connectivity in neural networks. https://opt-ml.org/papers/2022/paper90.pdf, 2022. Accessed: October 25, 2025.

Going beyond linear mode connectivity: The layerwise linear feature connectivity. In *NeurIPS*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/bf3ee5a5422b0e2a88b0c9c6ed3b6144-Paper-Conference.pdf. Accessed: October 25, 2025.

Model merging in llms & mllms: Methods and applications. https://www.emergentmind.com/papers/2408.07666, 2024. Accessed: October 25, 2025.

Proving linear mode connectivity of neural networks via optimal transport. In *Proceedings of Machine Learning Research*, 2024. URL https://proceedings.mlr.press/v238/ferbach24a/ferbach24a.pdf. Accessed: October 25, 2025.

An introduction to model merging for llms. https://developer.nvidia.com/blog/an-introduction-to-model-merging-for-llms/, 2024. Accessed: October 25, 2025.

Awesome model merging: Methods, theories, applications. https://github.com/EnnengYang/Awesome-Model-Merging-Methods-Theories-Applications, 2025. Accessed: October 25, 2025.

Slerp for model merging – a primer. https://www.coindeeds.ai/ai-blog/slerp-model-merging-primer, 2025. Accessed: October 25, 2025.

Training neural networks with symmetry constraints: A practical guide. https://medium.com/we-talk-data/training-neural-networks-with-symmetry-constraints-a-practical-guide-8f235ac9a469, 2025. Accessed: October 25, 2025.

Ainsworth, S., Hayase, J., et al. Git re-basin: Merging models modulo permutation symmetries. https://www.lesswrong.com/posts/4J8Cucvb5k7HHnkrL/git-re-basin-merging-models-modulo-permutation-symmetries, 2023. Accessed: October 25, 2025.

Anonymous. Procrustes alignment for model fusion. https://coindes.tech/procrustes/, 2024a. Accessed: October 25, 2025.

Anonymous. Rebasin: A new perspective on model merging. https://www.lesswrong.com/posts/bznrxT9e4mcXdvGPz/rebasin-a-new-perspective-on-model-merging, 2024b. Accessed: October 25, 2025.

Garipov, T., Izmailov, P., et al. Mode connectivity in loss landscapes of neural networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018. URL https://arxiv.org/abs/1802.10026. Accessed: October 25, 2025.

Kornblith, S., Norouzi, M., et al. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019. URL https://arxiv.org/abs/1905.00414. Accessed: October 25, 2025.

Singh, P., Chen, L., et al. Beyond permutations: Generalized symmetry alignment for model merging. https://arxiv.org/abs/2402.01862, 2024. Accessed: October 25, 2025.

Wen, Z. and Yin, W. Optimization algorithms on the stiefel manifold. *Mathematical Programming*, 2013. URL https://link.springer.com/article/10.1007/s10107-013-0682-1. Accessed: October 25, 2025.

Wortsman, M., Ilharco, G., et al. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without additional training. *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2203.05482. Accessed: October 25, 2025.