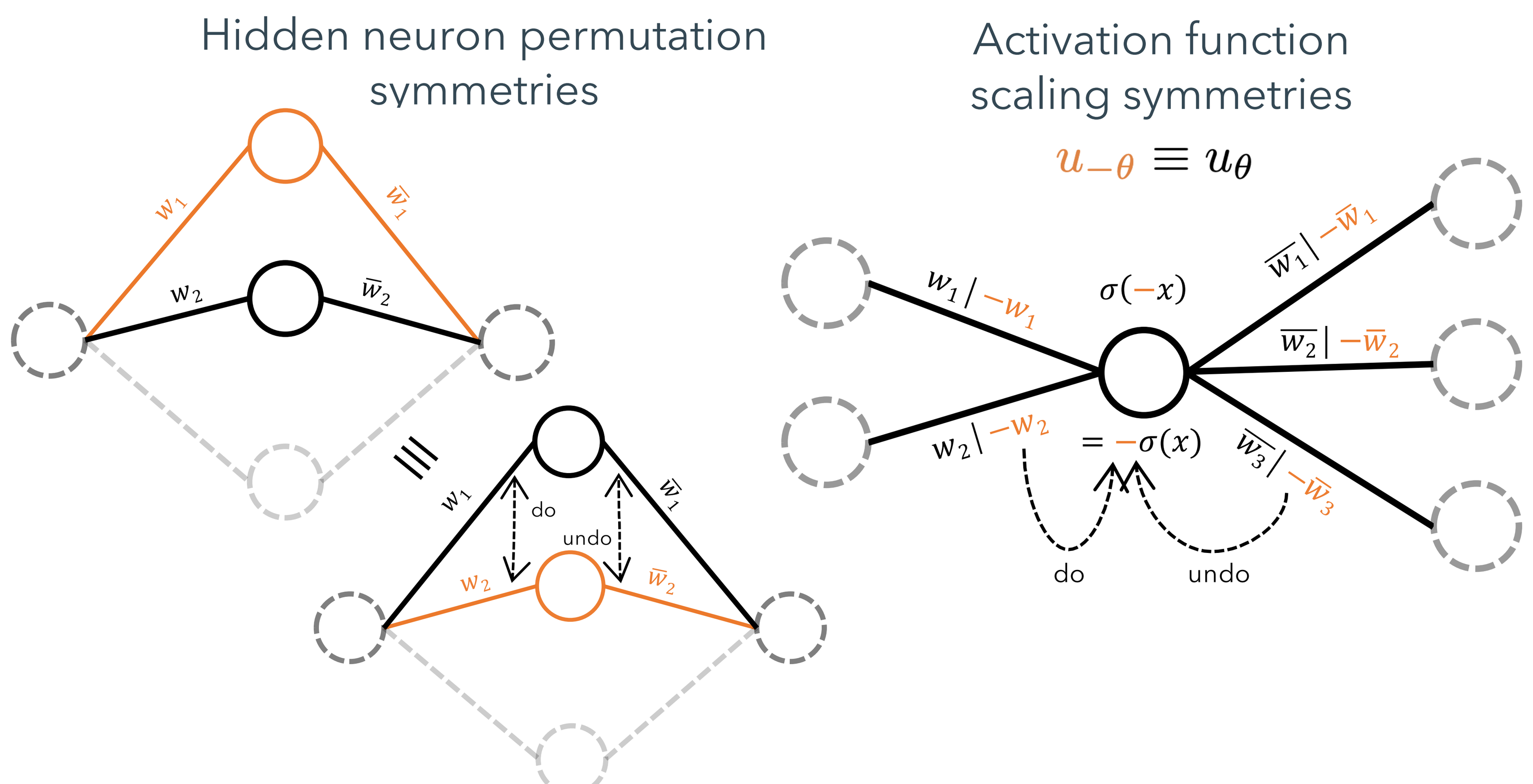
We present a method for **Model Merging through Parameter Canonicalization** using Symmetry-Aware Graph Metanetwork Autoencoders

Odysseas Boufalis
Jorge Carrasco Pollo
Joshua Rosenthal
Eduardo Terrés Caballero
Alejandro García Castellanos

Paper 19

1) Permutation and scaling symmetry groups in Neural Networks

Let $u_\theta(x)$ be a NN with parameters $\theta = (W_l,\ b_l)_{l=1}^{L}$.



i) Permutations:

$\theta_P = (P_l W_l P_{l-1}^{-1},\ P_l b_l)_{l=1}^{L}$

$P_l \in S_{d_l}$

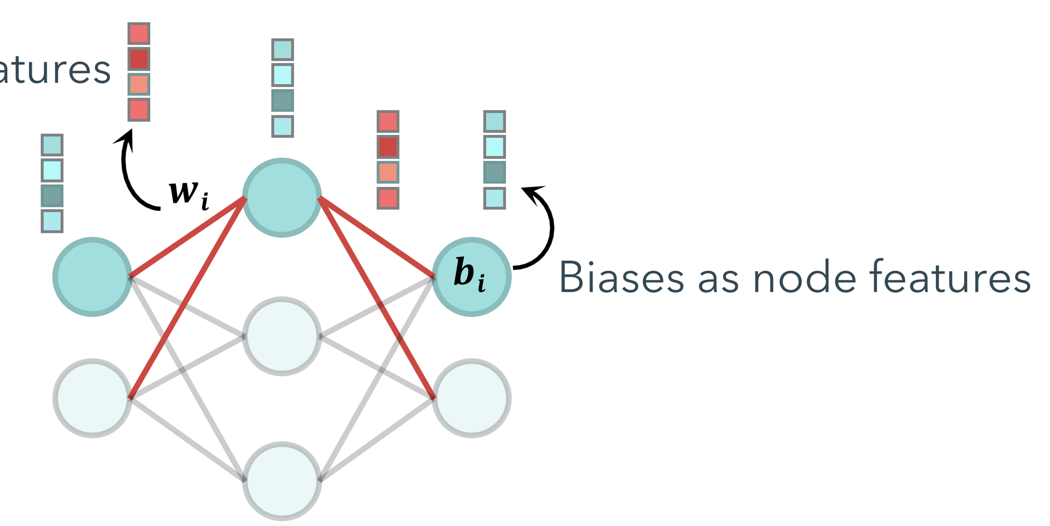
ii) Permutations + Scaling:

$\theta_{PS} = (P_l Q_l W_l Q_{l-1}^{-1} P_{l-1}^{-1},\ P_l Q_l b_l)_{l=1}^{L}$

$Q_l = \mathrm{diag}(q_1, \ldots, q_{d_l}),\ q_k = \pm 1$

2) The encoder: Symmetry-aware Graph Metanetworks

Metanetwork: A neural architecture designed to process the parameters and structure of other neural networks as its input.



Let $F: \Omega \to \mathbb{R}^D$ be the encoder.

i) Neural Graphs [1]

$$F(u_{G,\theta}(x)) = F(u_{G,\theta_P}(x))$$

Properties: Permutation equivariant message passing and permutation invariant readout.

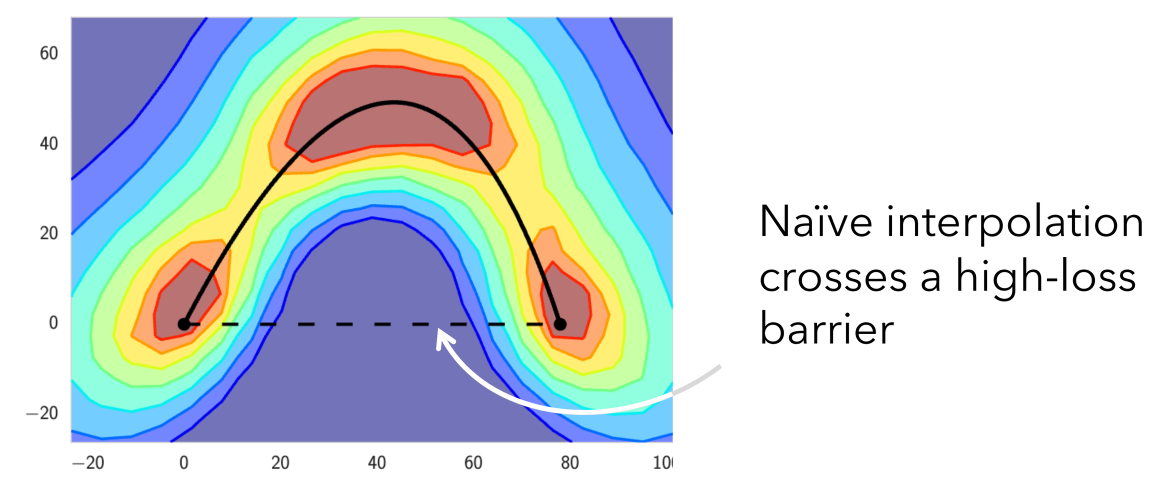
ii) ScaleGMNs [2]

$$F(u_{G,\theta}(x)) = F(u_{G,\theta_{PS}}(x))$$

Properties: Permutation and Scaling equivariant message passing and P+S invariant readout.

3) Model merging

Model merging combines networks through linear interpolation [4]. Symmetries create different **basins** in the loss landscape, where functionally equivalent networks lie.



Heatmap of the loss landscape w.r.t. model parameters. (Figure 4 from Garipov et al. [4])

Git Re-Basin [3]: Aligns networks by solving an assignment problem per layer via the Hungarian algorithm. Only corrects permutation mismatches.

We extend Git Re-Basin to also account for ± 1 scaling symmetries.

4) Canonicalization through autoencoding

The encoder maps all networks of a group orbit to the same latent vector. The decoder provides a learned canonical representation of this latent vector.

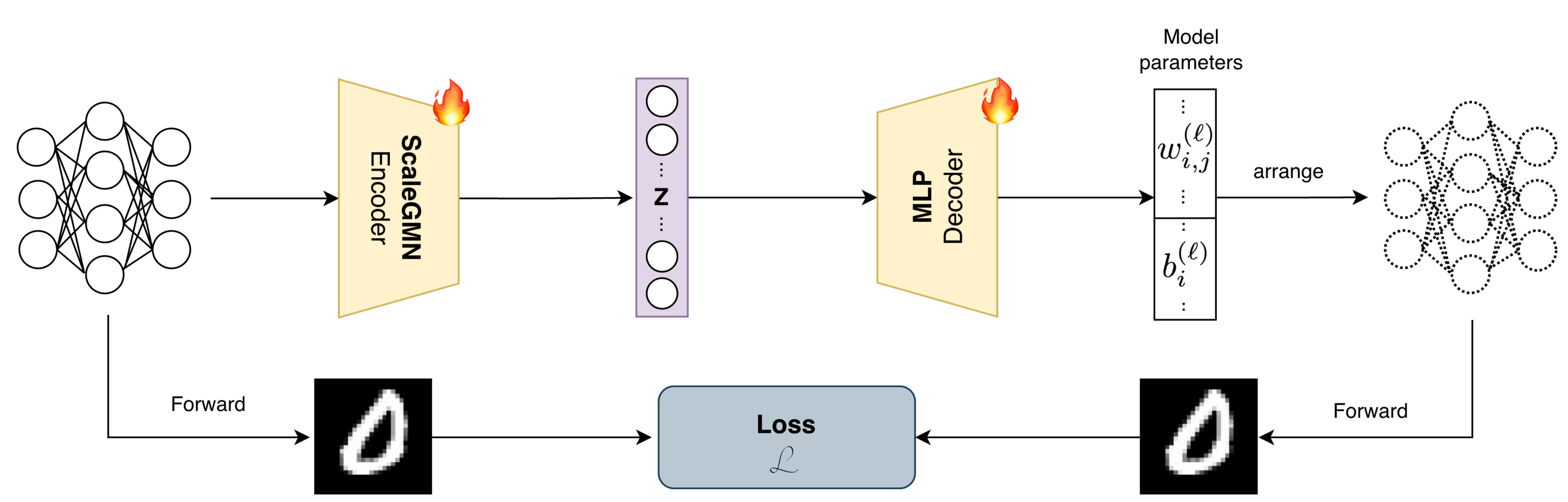


Figure: Autoencoder architecture

Autoencoder Loss: For INRs we minimize MSE on pixel activations to preserve the represented signal. For CNNs we minimize the KL divergence of the predicted class distributions after forward passing the image dataset.

Autoencoding offers linear computational cost but needs training. Linear assignment does not need training but is iterative and supra-linear in cost.

5) Results

Experiment 1: Interpolating MNIST INRs

We first perform the group action on an INR [5], then perturb the weights with Gaussian noise of variance ε (avoids encoding to equal latent vectors) and reconstruct the INR. We then perform linear interpolation in weight space.

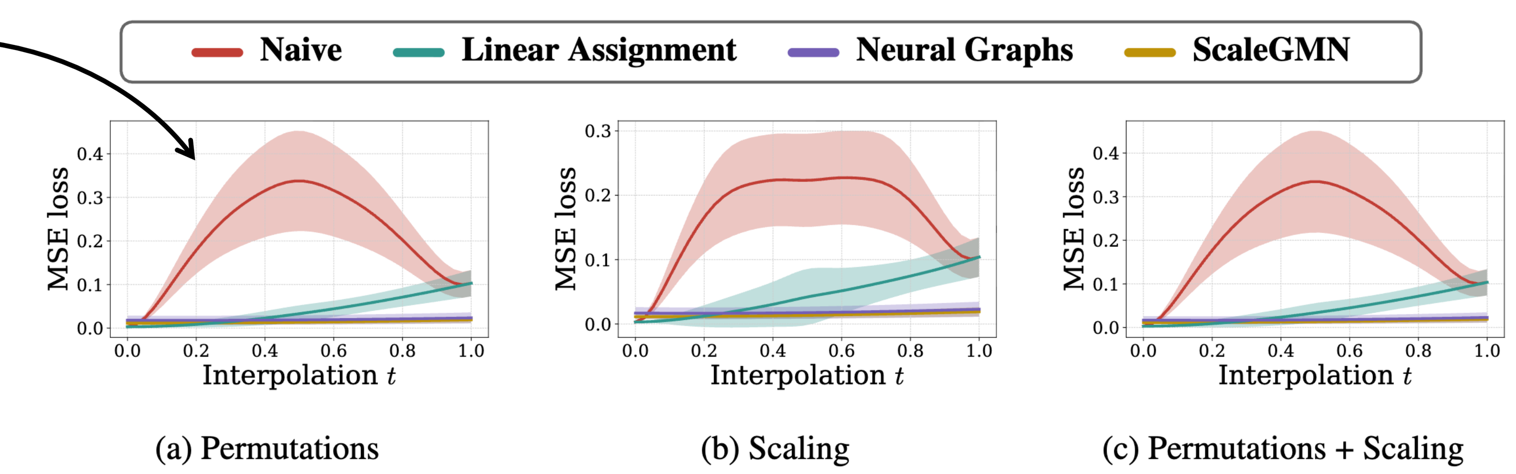


Figure: Network alignment on MNIST INRs and a group acted and perturbed version of them.

The interpolation given by the autoencoder presents a lower loss barrier and is robust to added noise in the weights.

Experiment 2: Interpolating CNNs

The CNNs [6] are trained on a fixed subset of CIFAR.

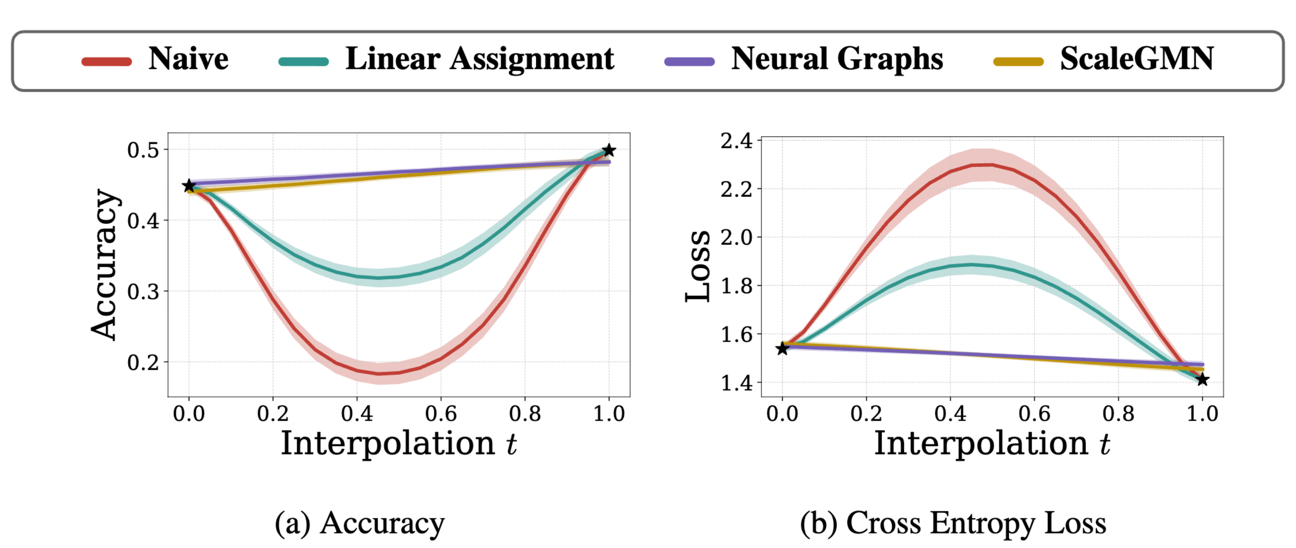


Figure: Network alignment on the highest performing CIFAR trained CNNs.

There is a tradeoff between lost accuracy after reconstruction and the better interpolation in terms of loss barrier.

References

[1] **Ioannis Kalogeropoulos et al.** Scale equivariant graph metanetworks. In: NeurIPS 2024.
[2] **Miltiadis Kofinas et al.** Graph neural networks for learning equivariant representations of neural networks. In: ICLR 2024.
[3] **Samuel K. Ainsworth et al.** Git re-basin: Merging models modulo permutation symmetries. In: ICLR 2023.
[4] **Timur Garipov et al.** Loss surfaces, mode connectivity, and fast ensembling of DNNs. In: NeurIPS 2018.
[5] **Aviv Navon et al.** Equivariant architectures for learning in deep weight spaces. In: ICML 2023.
[6] **Thomas Unterthiner et al.** Predicting neural network accuracy from weights. 2021.

Contact Information: a.garciacastellanos@uva.nl

University of Amsterdam