A. Related Work

Earlier works that operate on high-level statistics or the parameters of neural networks try to predict their final performance (Baker et al., 2017), hyper-parameters used for training (Eilertsen et al., 2020), and the generalization performance (Unterthiner et al., 2020). These works simply flatten the parameters and disregard the innate symmetry of neural network weights and biases. This is undesirable because models that are functionally identical (obtained by permuting the weights and biases appropriately) can receive vastly different predictions. A recent work performs different 3D tasks on implicit neural representations (De Luigi et al., 2023). They apply a set neural network on the set of vectors that includes all weight matrix rows and biases. This captures the wrong symmetries (the model is not equivariant/invariant to permuting the rows and columns of adjacent weight matrices as in equation 1 and 2), so they require a particular training method for the implicit neural representations to try to limit this issue.

A few recent works propose accounting for the permutation symmetry in neural networks by characterizing equivariant affine transformations on the parameter vector (Navon et al., 2023; Zhou et al., 2023). They do this via a complicated weight sharing pattern in the weight matrix of the affine transformation and show that this leads to significantly improved results compared to non-equivariant baselines. In contrast, in our work, we propose a graph-based representation for neural networks' weight spaces. This enables us to leverage the rich literature on graph neural networks, avoiding complications involving computing the weight sharing patterns of different neural network architectures or finding the right hyperparameters for the new architectures. This means that our method is more flexible: varying architectures are simply represented as different graph structures, which are easily handled by the graph neural network model. This flexibility also means that models with and without skip connections (which in principle have the same set of weights) can be distinguished. In contrast, the existing methods only support one architecture at a time because a different architecture requires changing the structured weights.