Generating Neural Networks from Equivariant Embeddings with Latent Diffusion

Saumya Gupta, Scott Biggs, Moritz Laber, Robin Walters, Ayan Paul
The Institute of Experiential AI, Northeastern University, Boston MA, USA
Network Science Institute, Northeastern University, Boston MA, USA
Khoury College of Computer Science, Northeastern University, Boston MA

Generating Neural Network Weight Matrices From Equivariant Embeddings with Latent Diffusion

MOTIVATION

Recently neural networks themselves have become the input of machine learning models. Applying generative models like diffusion models to generate neural network weight matrices is an especially promising direction for more efficient uncertainty quantification and transfer learning. However, the permutation symmetry and high dimensionality of weight matrices pose challenges for applying diffusion models to weight matrix generation. Here, we propose using latent diffusion for this task and combine different approaches to incorporate symmetry in our model. This includes using a permutation invariant auto-encoder to construct latent representations and choosing canonical representations for training.

Goals and Milestones

- We propose an architecture based on latent diffusion to efficiently generate weight matrices and biases for MLPs. This could be a promising route for transfer learning and uncertainty quantification.
- We explore the strengths and weaknesses of different approaches for incorporating weight space permutation symmetry in the auto-encoder of the latent diffusion model: Permutation equivariant linear layers and canonicalization.
- We find that the latent space structure reflects properties of the architectures in the training data. However, we find that maintaining MLP performance upon reconstruction poses a major challenge for our approach.

Weight Space Permutation Symmetry

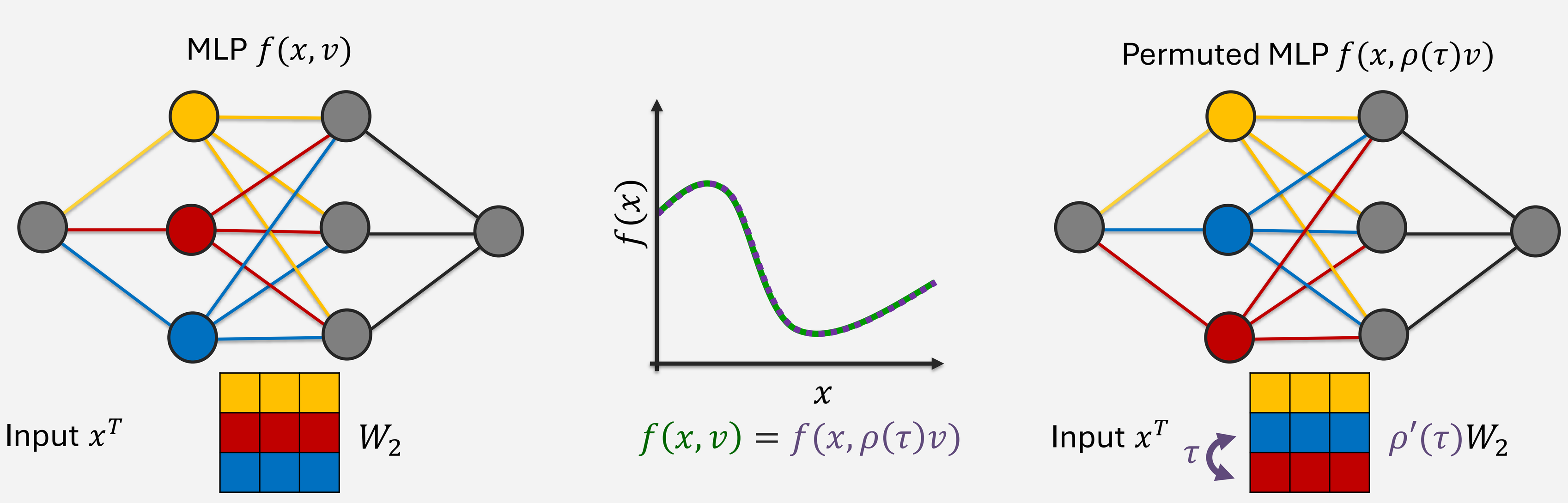

Fig.1: *Multilayer perceptrons (MLP) posses a permutation symmetry in weight space: Interchanging hidden neurons does not change the function implemented by the MLP. For weight space learning tasks, it can be beneficial to use architectures that are invariant or equivariant with respect to this symmetry.*

Weight Space Permutation Symmetry: A L-layer multilayer perceptron (MLP) is characterized by its weight matrices, $W_l \in \mathbb{R}^{d_l \times d_{l-1}}$ and biases $b_l \in \mathbb{R}^{d_l}$ at each layer $l \in [L]$. We represent them as elements of a joint vector space $\mathcal{V} = \oplus_{l \in L} W_l \oplus b_l$. The function implemented by the MLP does not depend on the ordering of hidden neurons, i.e., it is invariant under the action of the permutation groups $\mathrm{Sym}(d_l)$ at each l. Denoting $\rho(\tau = (\tau_1, \dots, \tau_L))$ the representation of $\tau \in \otimes_{l \in L} \mathrm{Sym}(d_l)$, this means

$$f(x,v) = f(x, v' = \rho(\tau)v) \text{ for all } \tau \in \otimes_{l \in L} \mathrm{Sym}(d_l),$$

with $\rho(\tau)$ transforming weights and biases as $W_l' = P_{\tau_l}^{\mathrm{T}} W_l P_{\tau_{l-1}}$ and $b_l' = P_{\tau_l}^{\mathrm{T}} b_l$ respectively. Navon et al. (ICML2023) design a set of equivariant linear layers, called *DWS layers*, that respect this symmetry and can be used for learning in weight space. We explore the potential of DWS layers as parts of an invariant auto-encoder architecture that creates low-dimensional latent representations of MLPs.

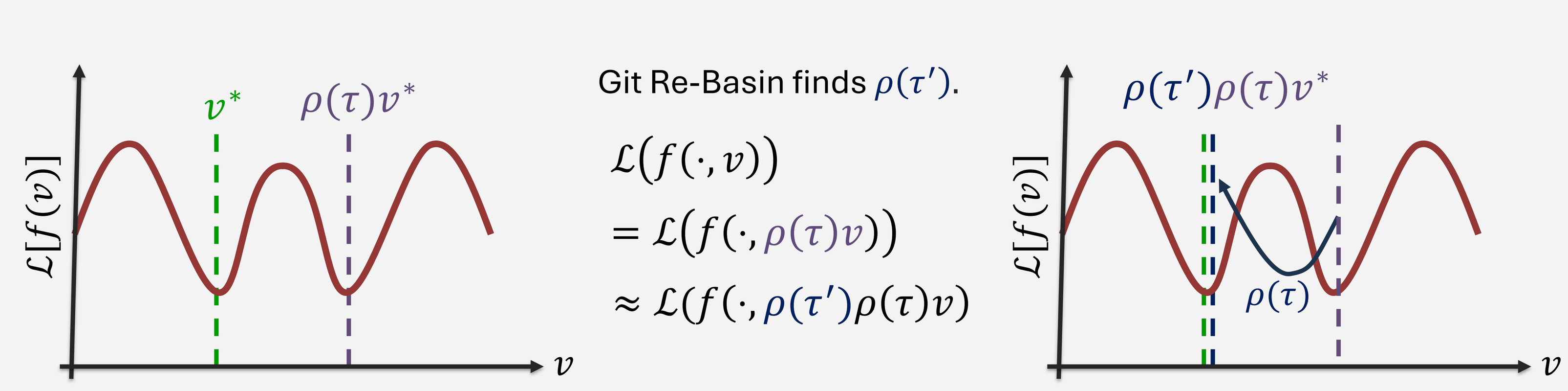

Fig. 2: *The invariance of MLPs with respect to permutation of hidden neurons is also reflected by symmetries of the loss landscape: If $f(x,v)$ and $f(x,\rho(\tau)v)$ implement the same function their respective loss values will be equal, $\mathcal{L}(f(\cdot,v)) = \mathcal{L}(f(\cdot,\rho(\tau)v))$, but correspond to different basins of the loss landscape. Tools from model merging, e.g., Git Re-Basin* by Ainsworth et al. (ICLR2025)*, can be used to approximately map different basins of the loss landscape to each other.*

Finding Mappings Basins of the Loss Landscape: The invariance of MLPs with respect to certain permutations of their weights, leads to a symmetries of the loss landscape: If the MLP implements the same functions for weights $v \in \mathcal{V}$ and $v' = \rho(\tau)v$, then the values of the loss function coincide, $\mathcal{L}(f(\cdot,v)) = \mathcal{L}(f(\cdot,v'))$. However, v and v' correspond to different, basins of the loss landscape. Establishing that there exists a permutation $\rho(\tau)$ that relates $v' = \rho(\tau)v$ and identifying is a NP-hard optimization problem but fast heuristics exist. Ainsworth et al. (ICLR2025) propose *Git Re-Basin* a heuristic to map MLPs that live in different basins but implement the same function to each other. We explore the potential of *Git Re-Basin* as part of the auto-encoder training process to identify a conical representative of the equivalence class under permutation. We use this canonical representation as one strategy to compute reconstruction loss of our auto-encoder.

Latent Diffusion in Weight Space

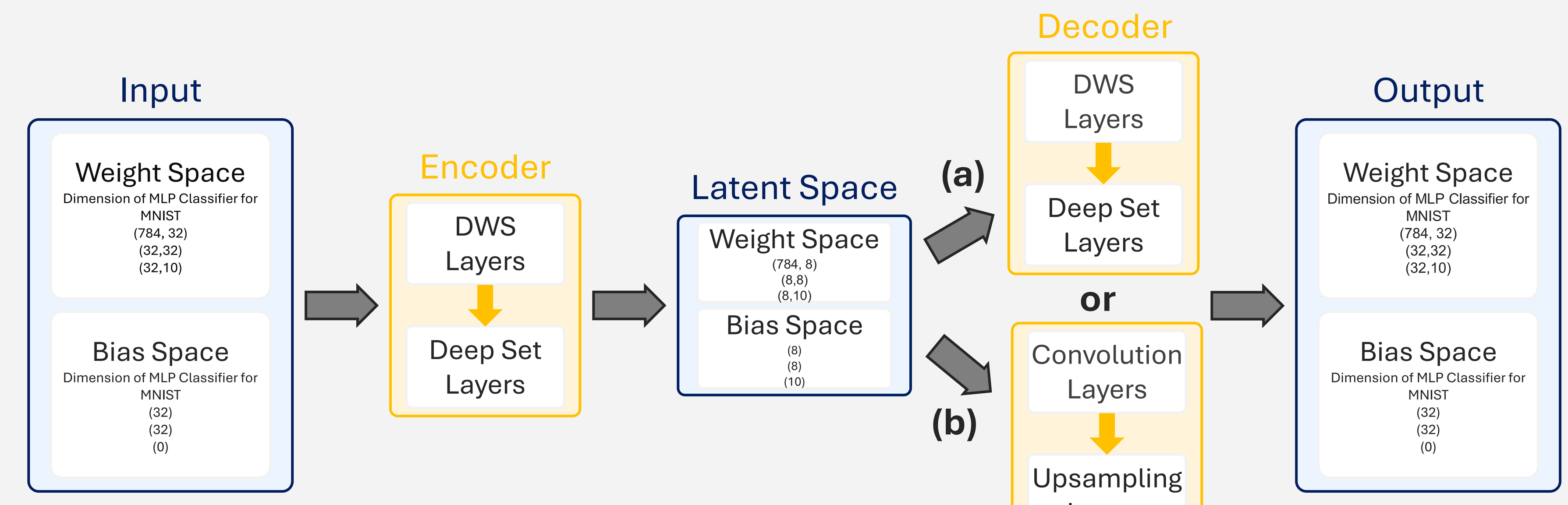

Fig. 3: *Permutation equivariant linear layers can be used to construct an auto-encoder that is invariant under the weight space permutation symmetry. This ensures that the auto-encoder maps weight vectors representing the same function to the same point in latent space. We use DWS layers (*Navon et al. ICML2023*) in the encoder to build such an auto-encoder. For the decoder we either mirror the encoder structure (path **(a)**) or use non-permutation-equivariant layers (path **(b)**).*

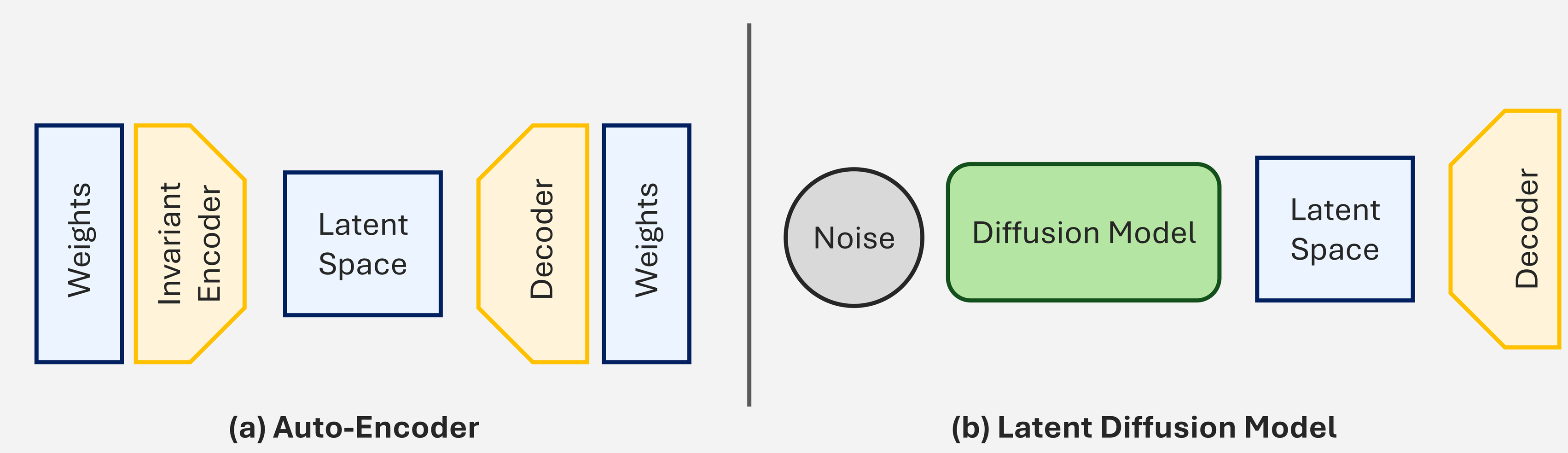

Fig. 4: *Latent Diffusion is a computationally efficient approach to generative modelling: Standard diffusion models rely on denoising diffusion process in a space of the same, and potentially high, dimensionality as the data. Latent diffusion models on the other hand, employ auto-encoders (panel **(a)**) to map the data into a low-dimensional latent space. The generative diffusion process takes place in this latent space (panel **(b)**). This makes latent diffusion models computationally more efficient.* (After Wang et al. *arXiv2402.13144*).

Preliminary Results

Latent Space Structure

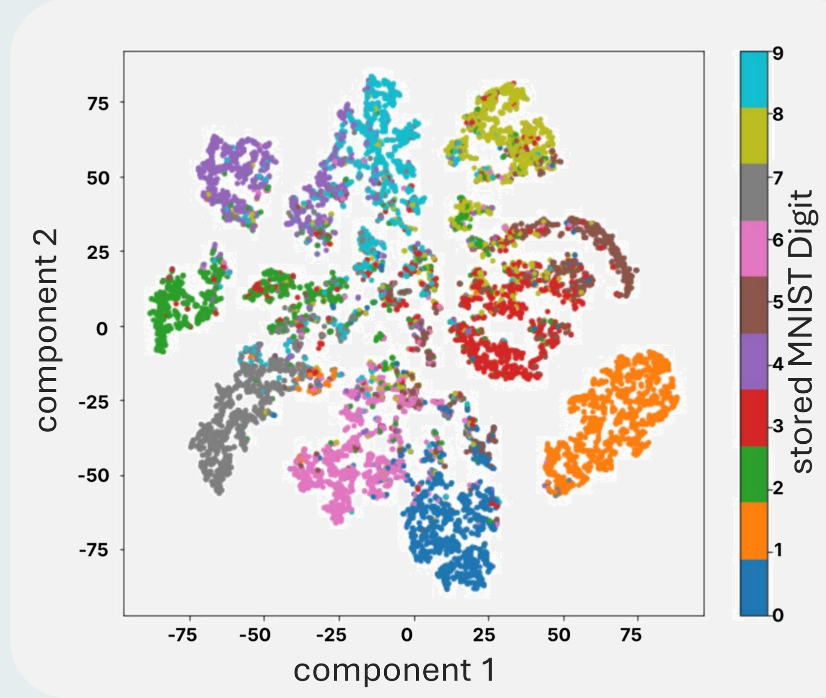

Fig. 4: *Low dimensional projection (obtained by TSNE) of the latent space of an invariant auto-encoder trained on INRs that store MNIST digits. The latent representations form clusters according to the stored digit.*

We trained a permutation invariant auto-encoder with DWS layers on implicit neural representations (INRs) of MNIST digits. We find that the latent space captures information about the task of the input network: In particular, we observe that the latent representations of INRs form clusters depending on which digit the INR stores. We visualize a low dimensional projection of the latent space, obtained by TSNE, in ***Fig . 4***.

Reconstruction Error

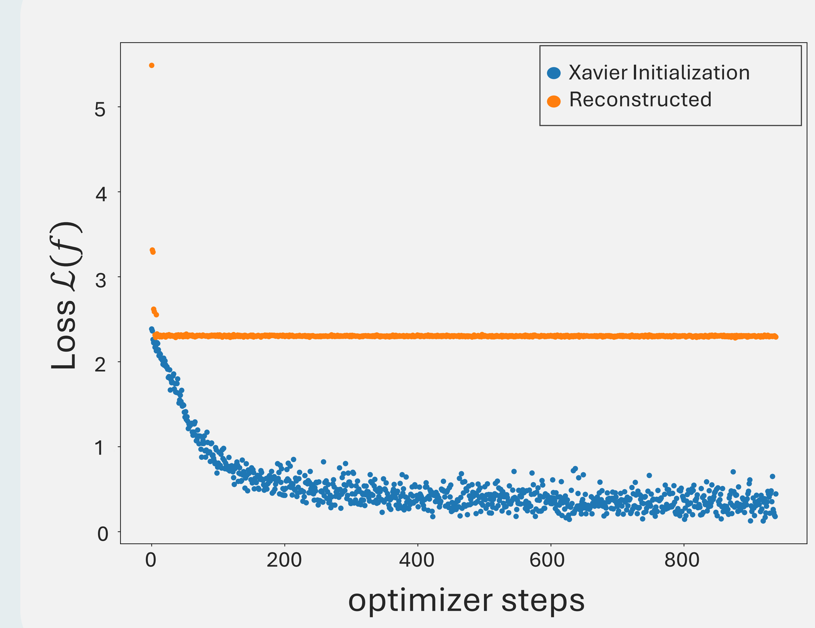

Fig. 5: *The performance of reconstructed MLPs is lower than of the encoded MLP. In comparison to MLPs initialized with Xavier initialization the reconstructed MLPs are also harder to (re)train.*

We examine the performance of MLPs reconstructed from the auto-encoders latent space on MNIST digit classification. As ***Fig. 5*** shows, reconstructing weights from latent representations is challenging: The reconstructed MLPs do not have low loss and are also hard to retrain. The loss for reconstructed MLPs is high initially and stagnates. MLPs initialized with Xavier initialization, on the other hand, show quickly decreasing loss on MNIST digit classification.

Summary

Applying latent diffusion models to learning in weight space is a promising direction for generative modeling of neural networks. Fast generation of neural networks from a low-dimensional latent space would enable more efficient transfer learning, model editing, and uncertainty quantification. However, weight space symmetries pose significant challenge to this approach. In this work we explore permutation equivariant layers and dataset canonicalization for obtaining latent representations of weight matrices.. We reveal that even though latent space structure captures important features of the encoded MLPs, reconstruction of high performing MLPs from their latent representation remains challenging. Moving forward, we plan to study the performance of generated MLPs in downstream tasks and applications of the full latent diffusion architecture to uncertainty quantification via ensemble methods and transfer learning via latent space interpolation.