SHERPA: LEVERAGING NEURON ALIGNMENT FOR KNOWLEDGE-PRESERVING FINE-TUNING

Dongkyu Cho[1], Jinseok Yang[1], Jun Seo[1], Seohui Bae[1], Dongwan Kang[1], Soyeon Park[1], Hyeokjun Choe[1], Woohyung Lim[1]
LG AI Research[1]

INTRODUCTION

Robust Fine-tuning

- Fine-tuning (FT) selected layers of a foundational model has shown great effectiveness in adaptation. However, the lack of clear criteria for layer-selection poses a significant obstacle. In this paper, we propose a novel approach to this problem by analyzing the loss landscape of trained networks.

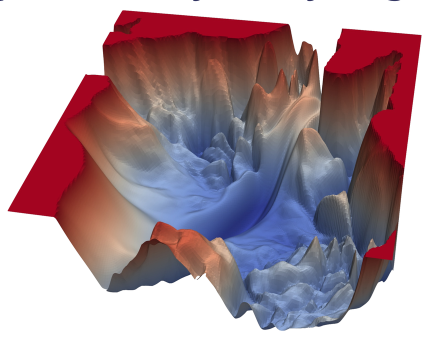


Figure 1. Loss landscape [1]

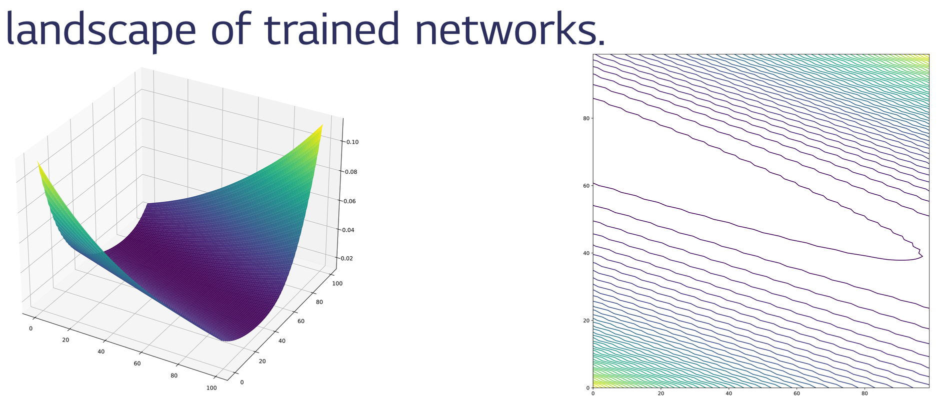


Figure 2. Loss landscape visualization (Left: surface, Right: contour)

Contribution

- We reveal that neuron alignment [2] can help preserve pre-trained knowledge amidst fine-tuning by exploiting the loss basin of trained models
- We present a 2 stage fine-tuning method ShERPA that enhances OOD generalizability without the additional cost of gradient computation
- We demonstrate that neuron alignment offers insights into how neural networks preserve and tune knowledge, revealing promising avenues for further exploration

PRELIMINARIES

Notation

- Let A: trained anchor model, M : training model, Θ_A: model weight of A, Θ_M: model weight of M.
- Let $\pi = (P_1, P_2 \cdots P_L)$ be a set of permutations that aligns L-layer networks A and $\pi(M)$ in their weight space.

Exploiting pre-trained models

- The robustness of foundational models derive from its pre-trained knowledge [6,7]. Fine-tuning the entire model inevitably distorts the pre-trained knowledge.
- Tuning only certain layers effectively boosts the OOD performance, while a reliable selection criteria is unknown [8].

Neuron Alignment

- In essence, neuron alignment algorithms align different models in their loss landscapes, leveraging the permutation invariance of neural networks [3].
- Neuron alignment is generally used to merge models in their weight space, such that individually trained models can be fused as one [4].

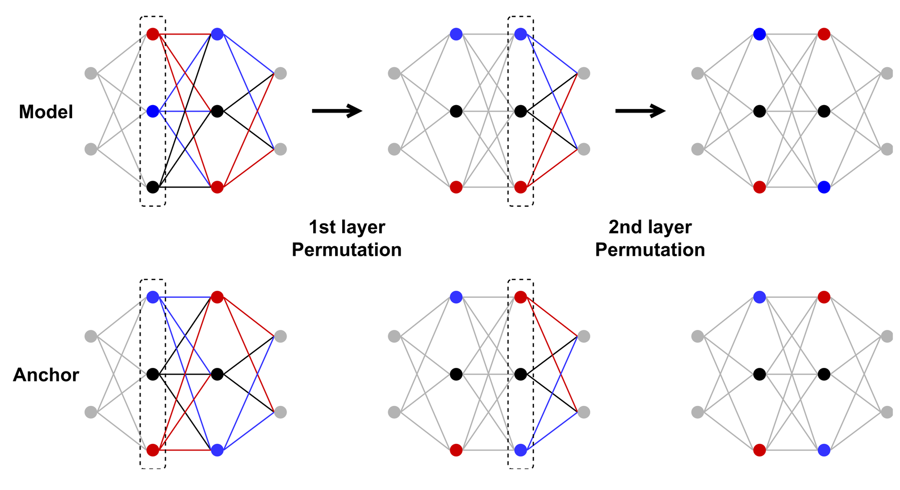


Figure 3. Neuron Alignment

MOTIVATION & PROPOSED METHOD

Motivation

- Fine-tuning a foundation model distorts pre-trained knowledge, damaging the model's robustness under distribution shifts [6]
- Models closely located in the same loss landscape share more pre-trained features [7]

Idea

- We use neuron alignment algorithms to shift the training model towards the basin of the trained anchor model in order to minimize the distortion of pre-trained knowledge.
- Analyze the difference between the original model M and the aligned model $\pi(M)$ to design a layer-selection criteria for parameter efficient fine-tuning.

Method

ShERPA (Shifted basin for Enhanced Robustness via Permuted Activations)

- Stage 1: Perform Neuron Alignment between A and M
- Stage 2: Fine-tune the neuron-aligned $\pi(M)$ on the source dataset.
- [Work-In-Progress] Stage 3: Analyze the aligned $\pi(M)$ for fine-tuning layer selection

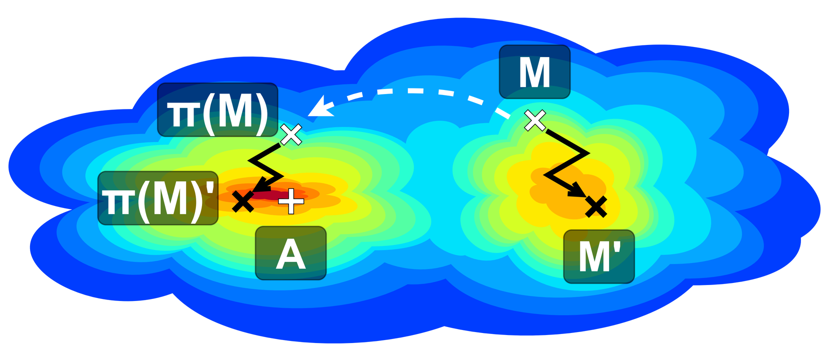


Figure 4. ShERPA framework

Our framework

Algorithm 2: ShERPA framework

Input: L-layer training model M and its weights Θ_M,
L-layer anchor model A and its weights θ_A, Data D,
fine-tune epochs n_{epochs}, permutation $\pi = (P_1, P_2, \cdots P_L)$;
Output: Trained Model $\pi(M)'$

Initialize A and M;
Pretrain A with D;

// Stage 1: Neuron-Alignment
for $l = 1 : L$ **do**
  Find l-th layer permutation P_l that minimizes Equation (1);
  Forward propagate the permutation P_l;
Apply the permutation set π to M;

// Stage 2: Fine-tuning
for $n = 1 : n_{epochs}$ **do**
  for $i = 1 : n_{iterations}$ **do**
    Sample i-th mini-batch from D;
    Forward and backward propagation of the mini-batch;
    Update $\pi(M)$;
return trained $\pi(M)'$

Alignment via activation matching

- A set of permutations π that aligns A and $\pi(M)$ minimizes:

$$\sum_i \mathrm{corr}\left(X^A_{(l,i)}, X^M_{(l,P_l(i))}\right), \quad (1)$$

for the i-th hidden unit in the l-th layer, where $X^A_{(l,i)}, X^M_{(l,P_l(i))}$ refers to the random variables representing the activations of the i-th hidden unit in the l-th layer.

- Optimizing Equation (1) maximizes the sum of correlations between the activations between A and M, which is a Linear Assignment Problem (LAP) that can be solved using combinatorial optimization methods [5].

Rationale for Neuron Alignment

- *Loss landscape of trained networks reflect their generalizability*
- *Alignment on the loss landscape will minimize knowledge distortion*

EXPERIMENT

Datasets

- Domain Generalization Benchmarks (e.g., PACS, Terra Incognita, VLCS)

Evaluation

- Fine-tune model on source domains, evaluate accuracy on OOD target domains.

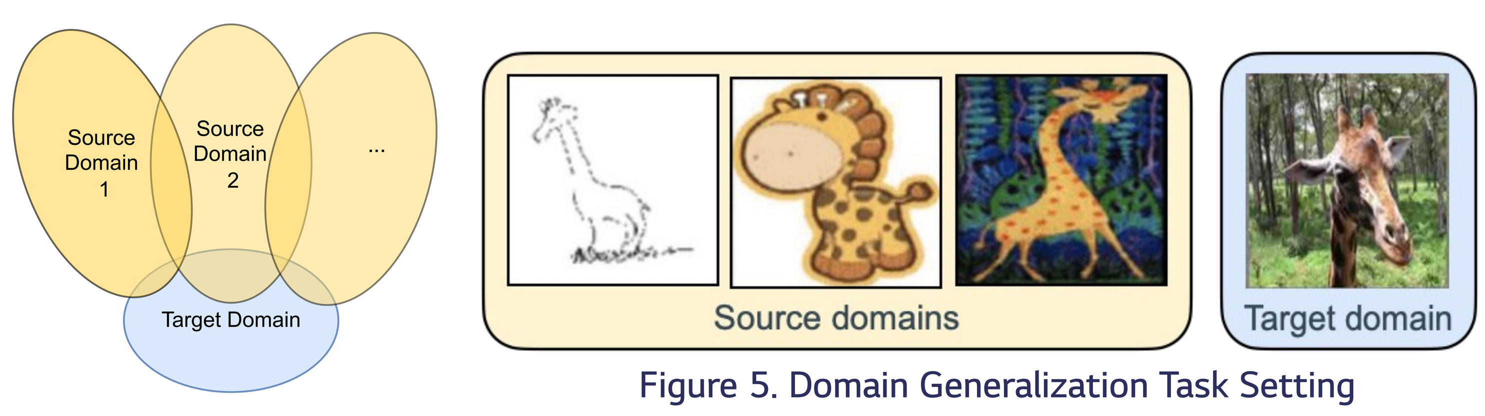


Figure 5. Domain Generalization Task Setting

ABLATION STUDY & FUTURE WORK

Study on Anchor

- We find that ShERPA's effects are not limited by the performance of the anchor A.

Neuron Alignment for layer selection

- We find potential in using neuron alignment to design a layer-selection criteria for parameter-efficient fine-tuning/ surgical fine-tuning.

REFERENCE

[1] Visualizing the Loss Landscape of Neural Nets (NIPS 2018)
[2] Convergent Learning: Do different neural networks learn the same representations? (NIPS 2015w)
[3] The Role of Permutation Invariance in Linear Mode Connectivity of Neural Networks (ICLR 2022)
[4] Model Fusion via Optimal Transport (NeurIPS 2020)
[5] A shortest augmenting path algorithm for dense and sparse linear assignment problems (DGOR/NSOR: Papers of the 16th Annual Meeting of DGOR in Co-operation with NSOR)
[6] Fine-tuning can Distort Pretrained Features and Underperform Out-of-Distribution (ICLR 2022)
[7] What is being transferred in transfer-learning? (NeurIPS 2020)
[8] Surgical Fine-tuning Improves Adaptation to Distribution Shifts (ICLR 2023)

QUANTITATIVE RESULT

Table 1: Accuracy on PACS.

Method	A	C	P	S	Avg.
ERM	91.22	80.63	98.03	67.32	84.3 ±0.2
Ensemble (m=6)	91.19	82.47	98.84	77.90	87.6
LP-FT (Kumar et al., 2022)	91.17	81.21	98.45	73.57	86.1 ±0.5
Random Perm.	87.80	84.64	97.85	71.06	85.3 ±**2.1**
SHERPA (Ours)	90.00	83.53	97.62	76.48	**86.9** ±0.1

Table 2: Accuracy on Terra Incognita.

Method	L100	L38	L43	L46	Avg.
ERM	61.11	40.15	48.54	40.00	47.4 ±0.4
Ensemble (m=6)	57.73	46.16	61.46	43.75	52.3
LP-FT (Kumar et al., 2022)	64.17	42.71	44.98	42.24	**48.5** ±0.5
Random Perm.	62.56	42.87	46.41	40.37	48.1 ±**0.7**
SHERPA (Ours)	64.63	41.28	45.47	41.78	**48.3 ±0.2**

Table 3: Accuracy on VLCS.

Method	C	L	S	V	Avg.
ERM	98.59	66.53	76.51	80.24	80.5 ±0.3
Ensemble(m=6)	98.02	66.11	78.55	81.61	81.0
LP-FT (Kumar et al., 2022)	99.08	67.10	76.44	80.58	**80.8** ±0.3
Random Perm.	97.40	63.00	72.50	76.30	77.3 ±**3.8**
SHERPA (Ours)	99.22	66.19	75.47	82.43	**80.8** ±0.2

Analysis on DG accuracy (Table 1,2,3)

- Neuron-Alignment boosts the target domain accuracy of fine-tuned models.
- Our framework (ShERPA) shows competitiveness against LP-FT, but falls behind an ensemble model.

Effect of neuron alignment on model parameters (Table 4)

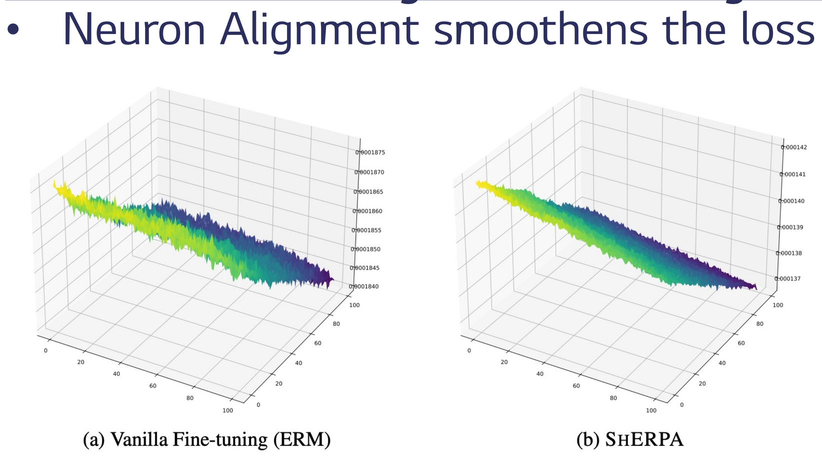
- Neuron Alignment keeps the model close in the parameter space

Table 4: ℓ_2 distance of ResNet-50 parameters before/after fine-tuning

Method	Conv1	Layer1	Layer2	Layer3	Layer4
Epochs=1					
ERM	0.0195	0.159	0.210	0.702	0.814
SHERPA (Ours)	**0.0159**	**0.127**	0.266	0.858	**0.674**
Epochs=10					
ERM	0.0395	0.282	0.631	2.235	2.257
SHERPA (Ours)	**0.0263**	0.289	0.669	**2.125**	**2.006**
Epochs=30					
ERM	0.0333	0.367	0.753	3.776	2.696
SHERPA (Ours)	**0.0293**	**0.343**	1.141	**3.736**	**2.417**

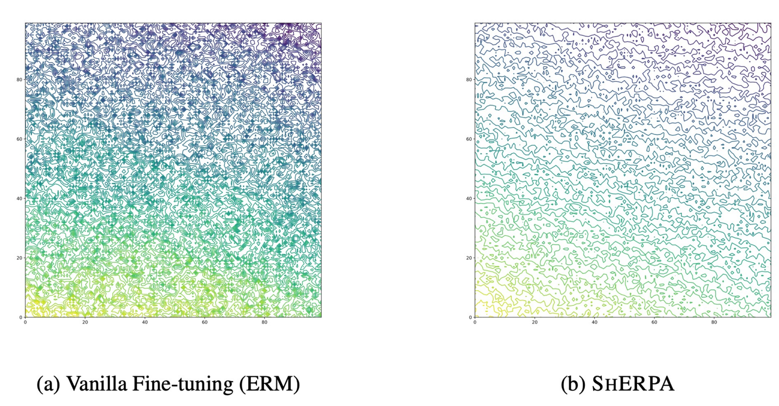
Effect of neuron alignment on loss geometry

- Neuron Alignment smoothens the loss surface (Below)



(a) Vanilla Fine-tuning (ERM) (b) SHERPA

Figure 2: The loss surface of trained models



(a) Vanilla Fine-tuning (ERM) (b) SHERPA

Figure 3: The loss contour of trained models