Winning the Initialization Lottery:

Towards Faster Training and Better Performance through Weight Initialization

Diego Aguirre, Ph.D.

Progress Report: 06/12/22

Team Motivation and Objectives

- The number of parameters in models is growing (almost) exponentially.
  - Need more compute, more data, more time, more money, etc.
- We want self-supervised models that:
  - Are not unnecessarily large
  - Perform well
  - Have a reasonable inference time

The Importance of Weight Initialization

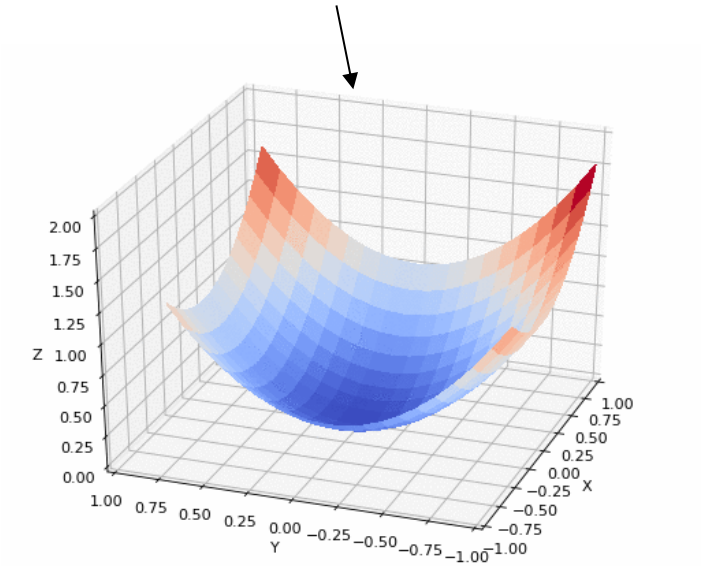
Weight Initialization: Why?

We know* that:

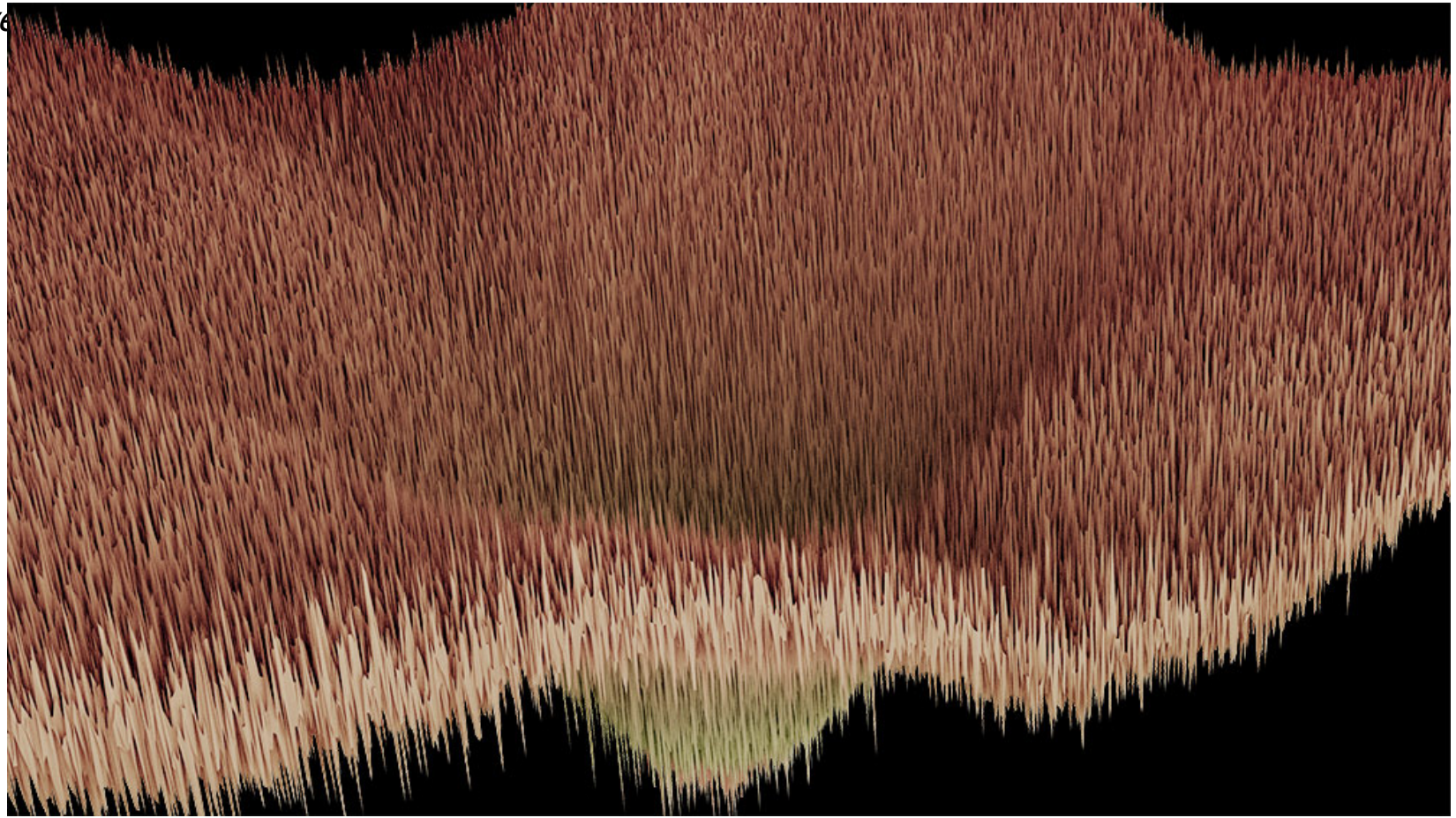
- Large models can be compressed
- “Lottery tickets” trained in isolation perform almost as well as the original network

Popular solution: buy as many tickets as you can afford

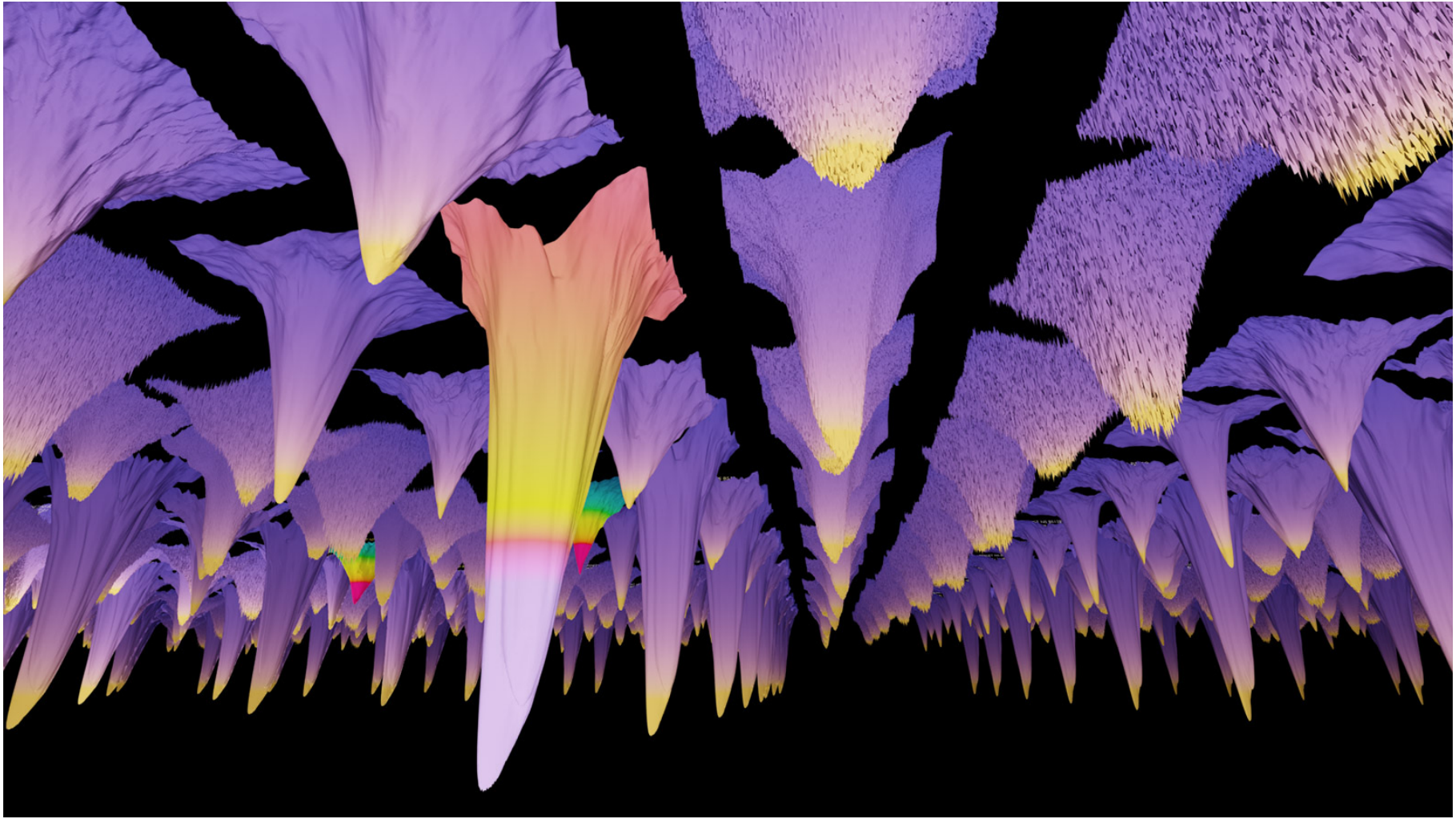
Ideal loss landscape



As networks become larger, the loss landscape is more likely to become *chaotic* and *highly non-conve*

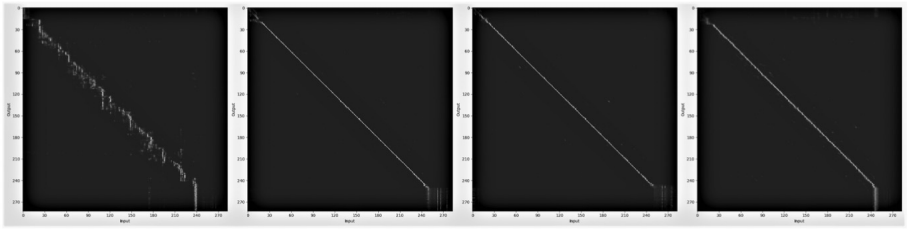


https://losslandscape.com/

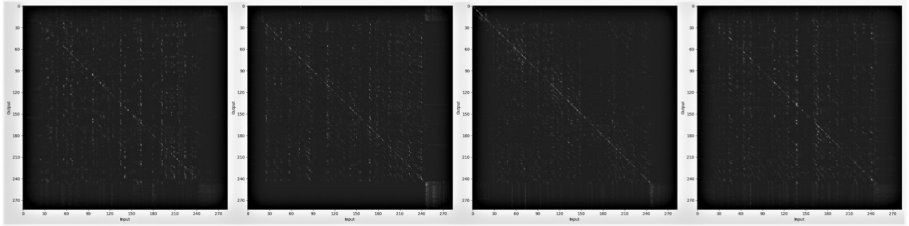


https://losslandscape.com/

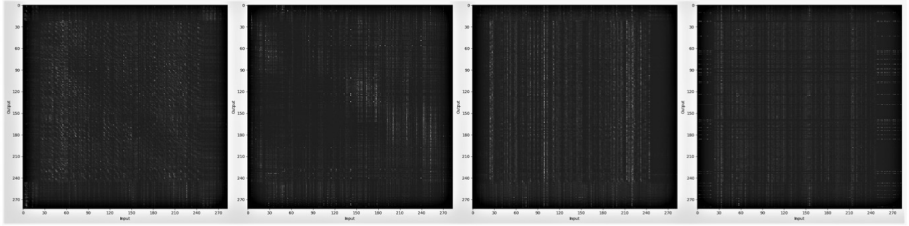
What do Transformers learn?



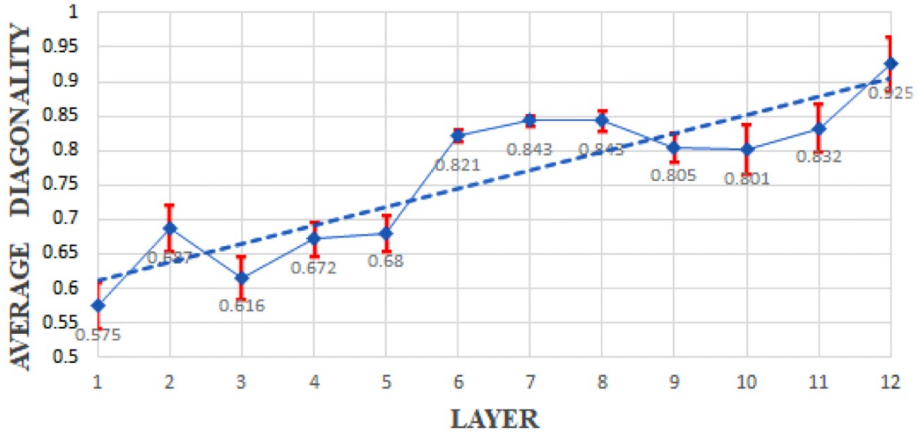
(a) Attention vectors of each attention head of encoder self-attention layer 12



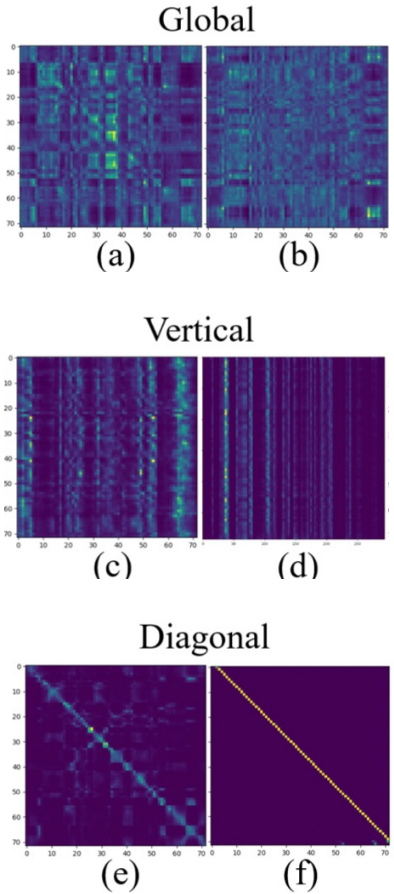
(b) Attention vectors of each attention head of encoder self-attention layer 5

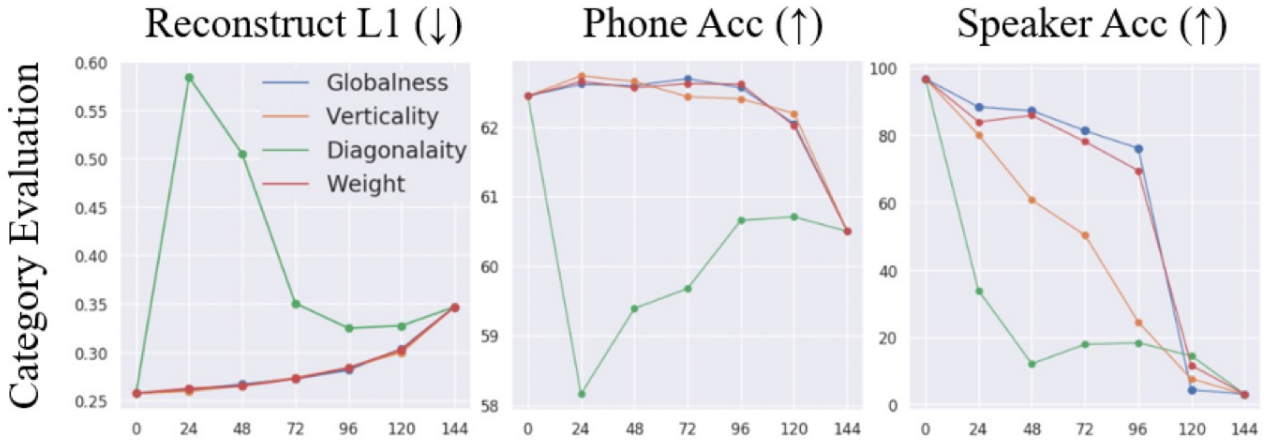


(c) Attention vectors of each attention head of encoder self-attention layer 1



[Zhang, Shucong, Erfan Loweimi, Peter Bell, and Steve Renals. "On the usefulness of self-attention for automatic speech recognition with transformers." In 2021 IEEE Spoken Language Technology Workshop (SLT), pp. 89-96. IEEE, 2021.]

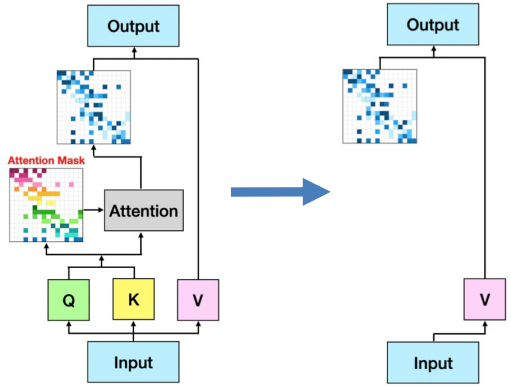




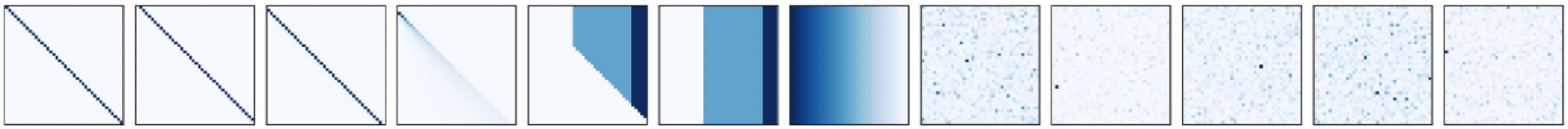
Key findings from the authors:

(1) Diagonal heads are the most important.
(2) Vertical heads rank second
(3) Global heads have the least importance
(4) Both global and vertical heads are harmful to the phonetic structure

[Yang, Shu-wen, Andy T. Liu, and Hung-yi Lee. "Understanding self-attention of self-supervised audio transformers." arXiv preprint arXiv:2006.03265 (2020).]



(a) Attention weights we used to initialize our model. The five leftmost weights are diagonal, shifting diagonal left one step, two steps, and shifting diagonal right one step, two steps. The five rightmost weights are initialized with small random numbers. As for the middle two weights, we initialize them using the numbers increasing from left to right and decreasing from left to right respectively.

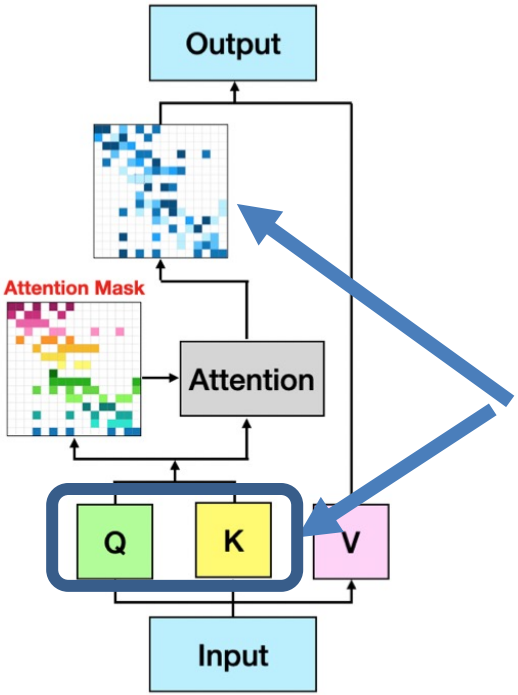


(b) Attention weights learned by our model after pre-training.

[Wu, Tsung-Han, Chun-Chen Hsieh, Yen-Hao Chen, Po-Han Chi, and Hung-yi Lee. "Input-independent Attention Weights Are Expressive Enough: A Study of Attention in Self-supervised Audio Transformers." arXiv preprint arXiv:2006.05174 (2020).]

Weight Initialization

Transformer Initialization



Use a subset of the training dataset to initialize weight matrices of Q and K so the resulting attention maps have desirable properties (e.g. highly diagonal)

Current Implementation

1. Initialize Wq, Wk, Wv matrices of all attention heads in an attention layer using orthonormal vectors

2. Feed initialization set and solve for Wq using pseudo inverse

(Emb * Wq) * K.T /sqrt(dim) = logits of desired attention scores

(Emb * Wq) * K.T = logits of desired attention scores * sqrt(dim)

(Emb * Wq) * K.T = L

Emb * Wq = L * pseudo_inv(K.T)

Wq = pseudo_inv(Emb) * L * pseudo_inv(K.T)

k	-k	-k	-k
-k	k	-k	-k
-k	-k	k	-k
-k	-k	-k	k

Initial Experiments

Pre-training using S3PRL codebase

Upstream Model: Mockingjay

Data: [train-clean-100, train-clean-360, train-other-500]

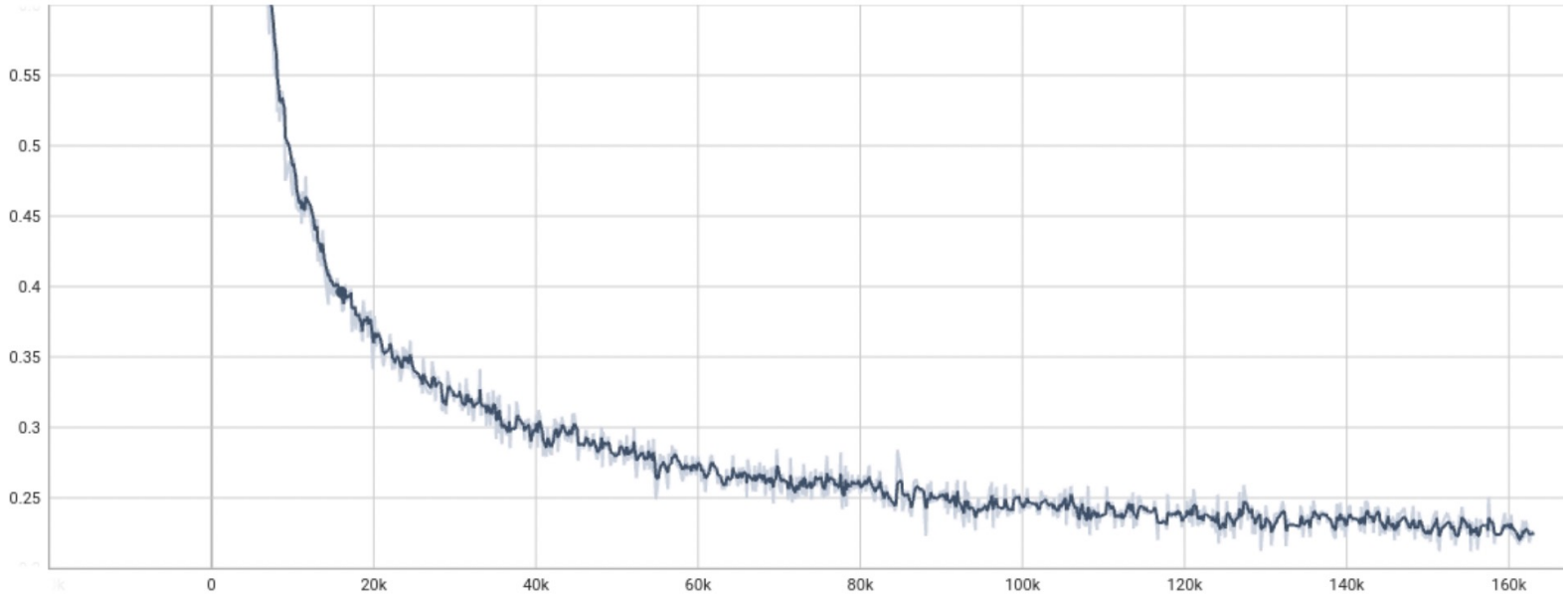
GPU: RTX 3090

Training Batch Size: 9

Optimizer: AdamW w/ scheduler – lr: 5.e-5

Initial Experiments – Default Initialization

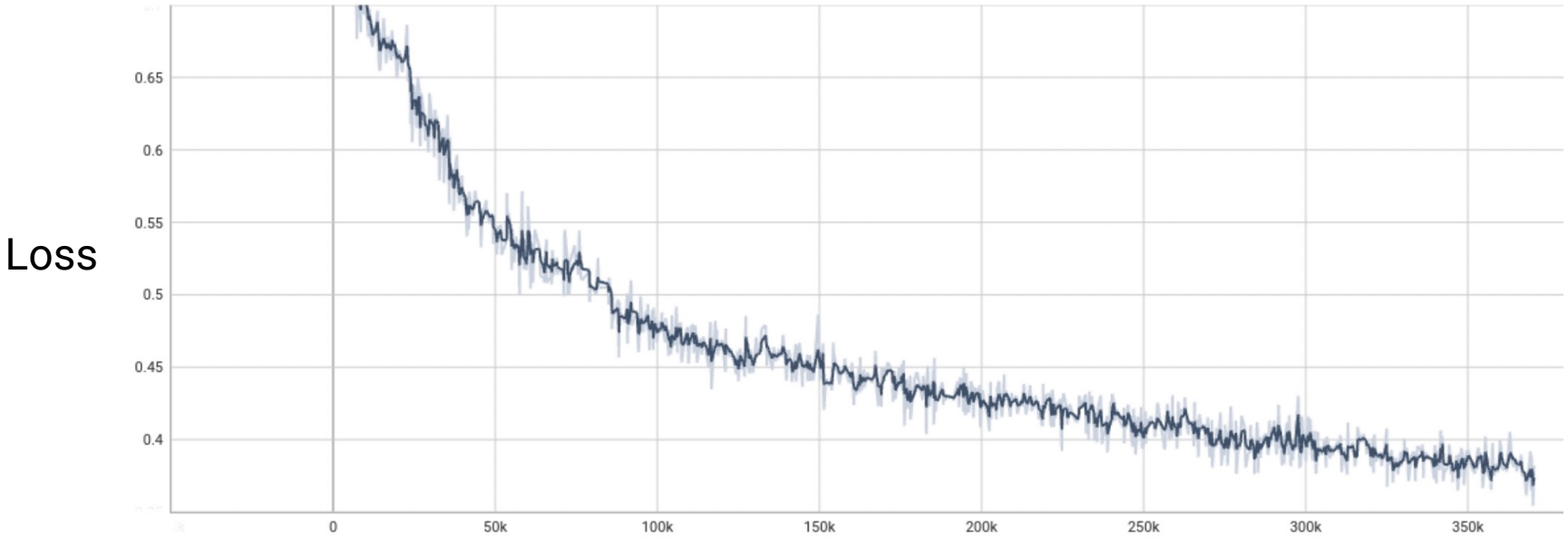
Loss



Training Steps

Loss (step 150k) = ~0.221

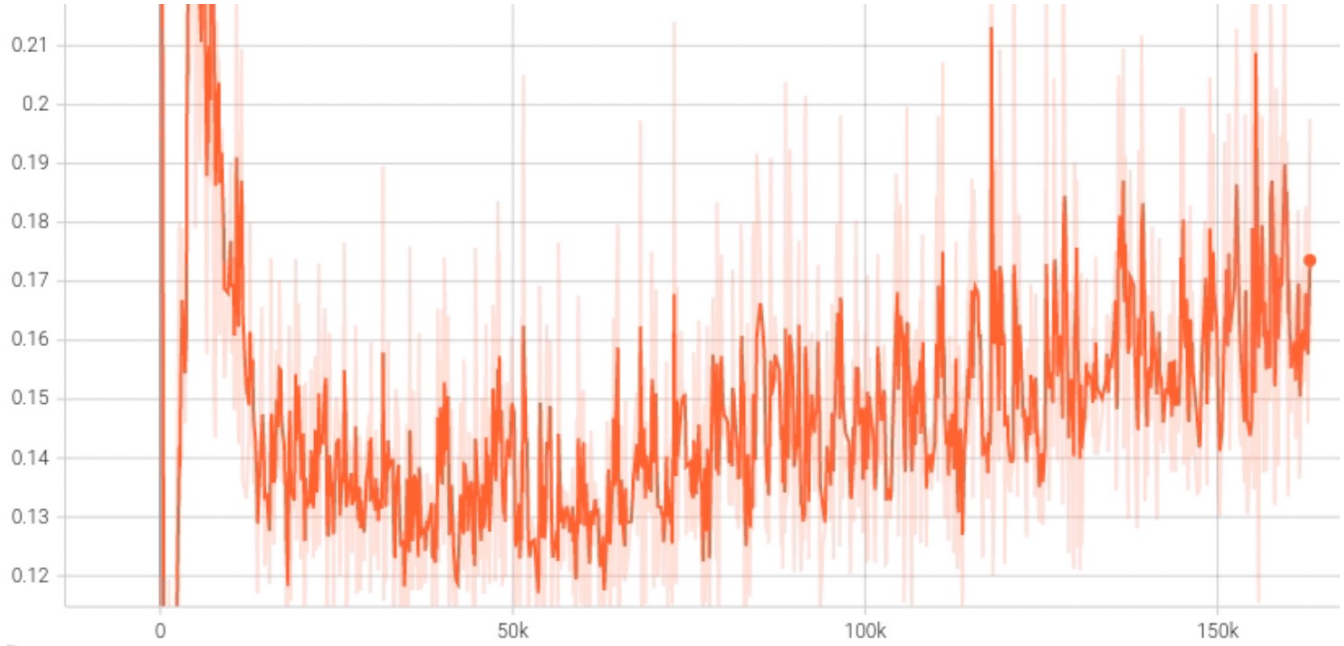
Initial Experiments – Proposed Initialization



Loss (step 350k) = ~0.384

Initial Experiments – Default Initialization

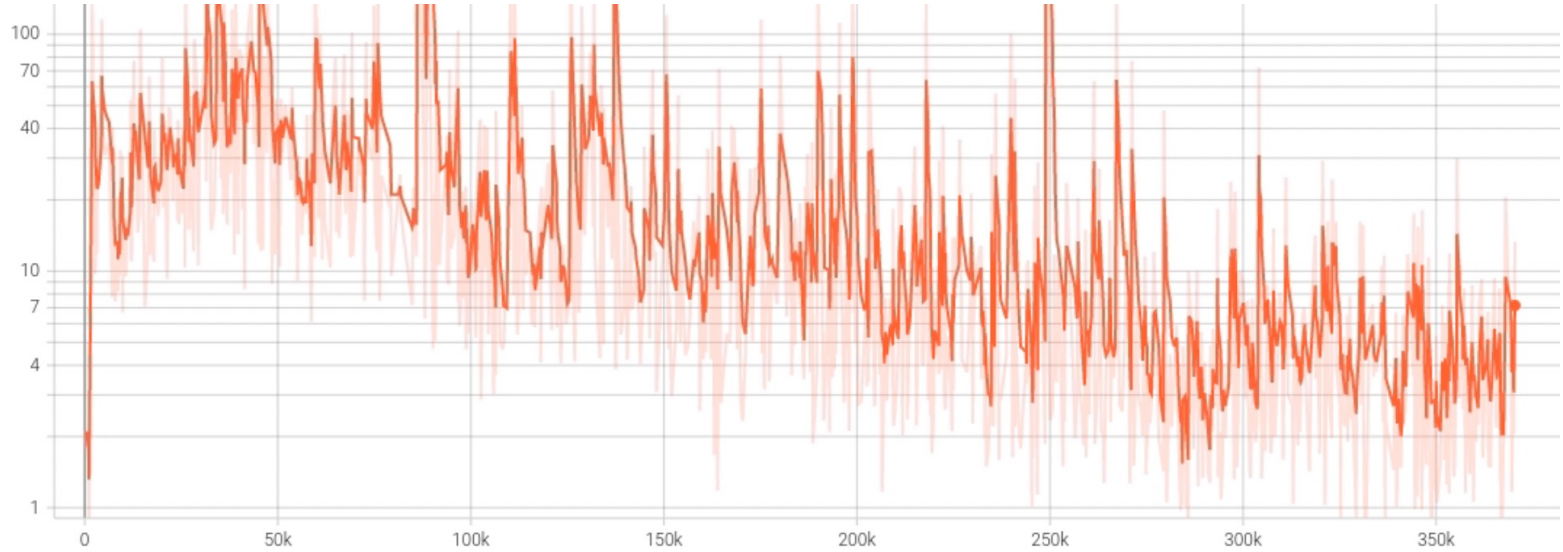
Gradient Norm



Training Steps

Initial Experiments – Proposed Initialization

Gradient Norm



Training Steps

Initial Experiments – Preliminary Results

Downstream Task: SUPERB phone_linear

Default S3PRL configuration, except for number of steps (set to 50k)

Initialization	Checkpoint Step	Pre-training loss	Best dev acc	Best test acc
Default	150k	0.221	0.577	0.578
Proposed	150k	0.440	0.597	0.599
Proposed	350k	0.384	0.623	0.624

Plan / Timeline

Before the end of the first week of the workshop

Add initialization logic to Mel-HuBERT

Train: Libri-360 – Test: Phone classification on WSJ dataset

During the workshop (before the end of July)

Improve initialization approach

- Use a linear SVM instead of pseudo-inverse?

- Relax diagonalization requirements? Add others?

Train: Libri-960 – Test: SUPERB tasks

Questions?

Thank you!

Diego Aguirre

daguirre6@utep.edu