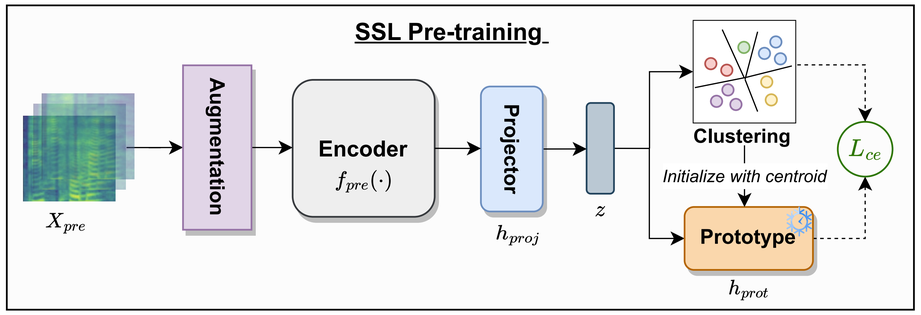
UNFUSED: Unsupervised Finetuning Using Self-supervised Distillation

Ashish Seth[1*], Sreyan Ghosh[2*], S.Umesh[1], Dinesh Manocha[2]
[1]Indian Institute of Technology, Madras, [2]University of Maryland, College Park

Motivation: Unsupervised Finetuning and Model Compression

- We introduce UnFuSeD (Unsupervised Finetuning Using Self-supervised Distillation), which does unsupervised finetuning using SSL pre-trained models.
- UnFuSeD also achieves model compression of **40%** using self-distillation by dividing the encoder into student and teacher counterparts.

Proposed Upstream SSL pre-training: DECAR-v2



We propose DECAR-v2 which stablize the overall training process by updating the Prototype weights with cluster centroids.

- **Assignment Phase**: The primary purpose of this phase is to obtain pseudo-labels $\mathbf{q}$ for every unlabelled audio sample $x \in X_{pre}$
- **Training Phase**: We train the network using supervision from the pseudo-labels $\mathbf{q}$ obtained from the assignment phase.

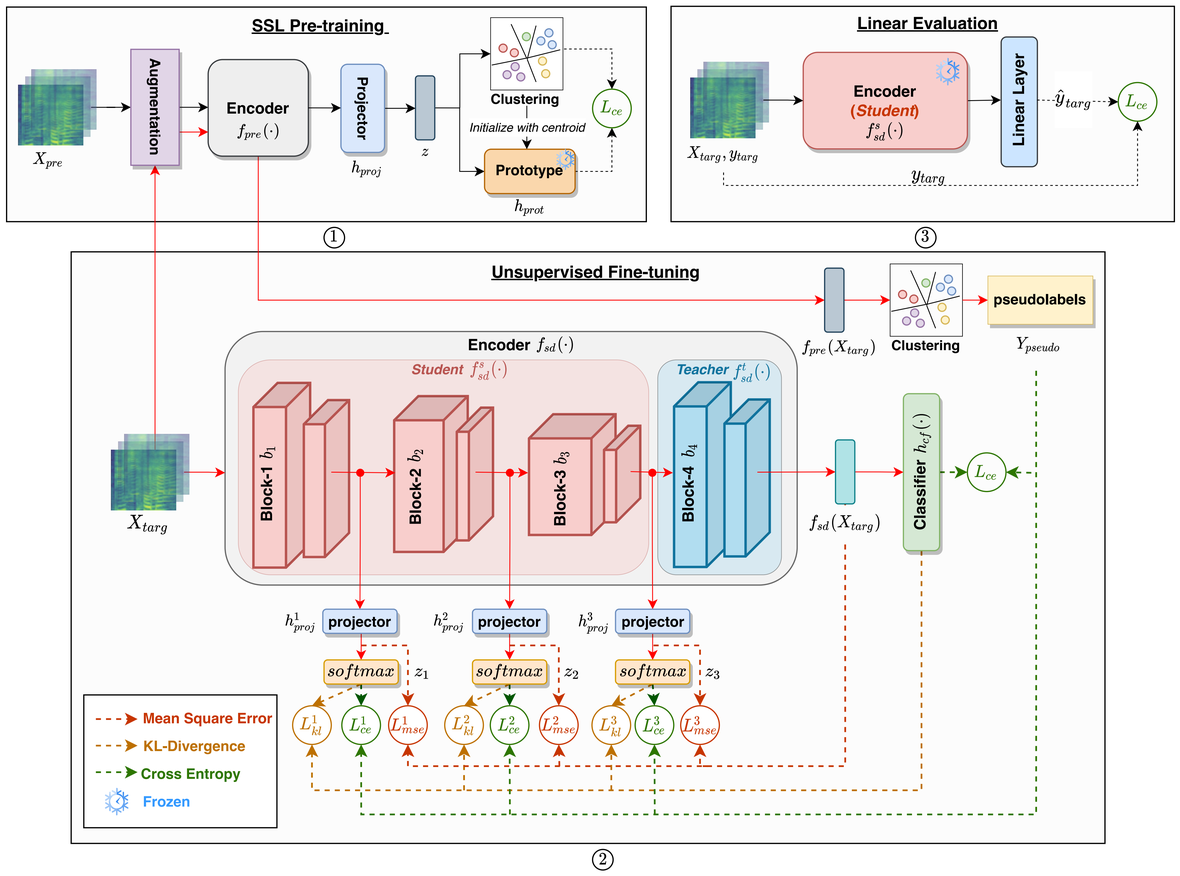
Results: Comparing SLICER performance

Models have been pre-trained on 10% of AudioSet and FSD50K and then linearly evaluated while keeping the weights frozen on the LAPE benchmark.

Model	Speech Tasks							Non-Speech Tasks			
	SC-V1	SC-V2 (12)	SC-V2 (35)	LBS	VC	IC	VF	NS	BSD	TUT	US8K
COLA	77.3	77.2	66.0	89.0	28.9	59.8	69.2	61.3	85.2	52.4	69.1
BYOL	87.7	87.2	84.5	90.0	31.0	60.0	83.1	71.2	87.8	58.4	77.0
DeLoRes-S	86.1	85.4	80.0	90.0	31.2	60.7	76.5	66.3	86.7	58.6	71.2
DeLoRes-M	94.0	93.3	89.7	95.7	45.3	65.2	88.0	75.0	89.6	65.7	82.7
UnFuSeD	**94.4**	**94.1**	**90.1**	**97.0**	**50.0**	**66.0**	**89.8**	**76.4**	**90.0**	**66.8**	**83.2**

UnFuSed achieves SOTA performance with an average gain of **1.2%** and **40%** fewer parameters compared to DeLoRes-M.

Proposed Architecture for UnFuSeD



Unsupervised Finetuning: We finetune a randomly initialized encoder with the pseudo-labels obtained by a pre-trained encoder using self-distillation by dividing the encoder $f_{sd}(\cdot)$ into the student $f^s_{sd}(\cdot)$ and teacher $f^t_{sd}(\cdot)$ counterparts.

Loss Functions: We jointly optimize Cross-Entropy, KL-divergence and Mean-square error across student and teacher counter part.

$$L_{all} = L_{ce} + \alpha \sum_{i=1}^{3} L^i_{ce} + (1-\alpha) \sum_{i=1}^{3} L^i_{kl} + \beta \sum_{i=1}^{3} L^i_{mse}$$

Linear Evaluation: Next, we use only the student counterpart $f^s_{sd}(\cdot)$ and update a linear layer on the desired downstream tasks while keeping the rest of the network frozen.

Paper Code