



DeepCluster

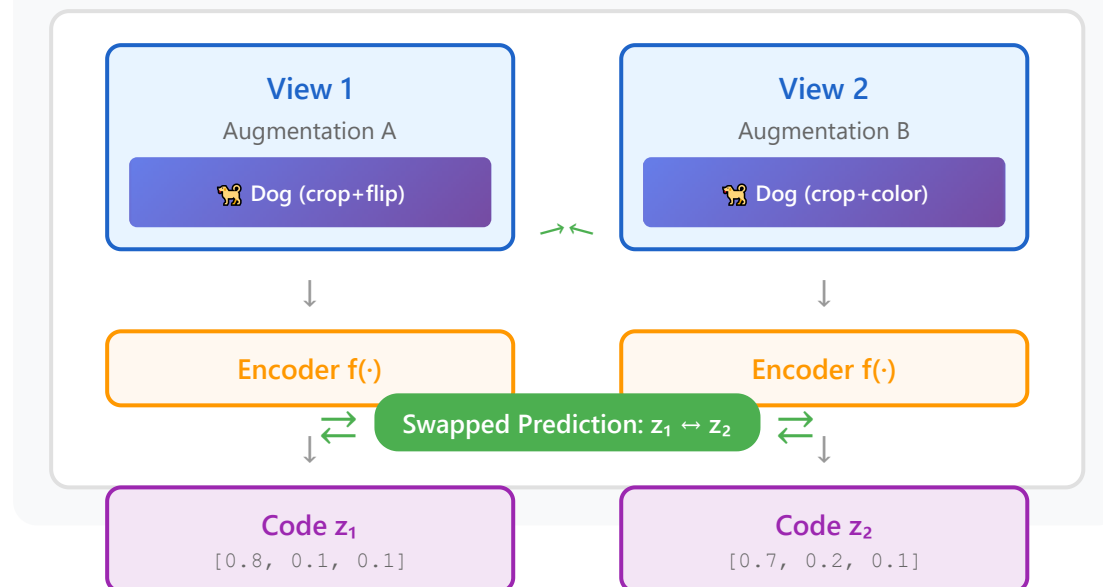
- ✓ Alternates clustering and representation learning
- ✓ Cluster assignments as pseudo-labels
- ✓ Iterative refinement process



SwAV

- ✓ Swapping Assignments between Views
- ✓ Online clustering with prototypes
- ✓ Learnable cluster centroids
- ✓ No pairwise comparisons needed

SwAV Architecture & Training Process



Efficient

No pairwise comparisons



Online

Real-time clustering



Prototypes

Learnable centroids



Scalable

Large datasets



State-of-the-Art Self-Supervised Learning Results



DeepCluster Operating Principle

Algorithm Flow

1

Feature Extraction

Transform images into feature vectors using CNN

$$x \rightarrow f(x) \in \mathbb{R}^d$$

2

K-means Clustering

Group features into K clusters

$$\text{assign: } \operatorname{argmin} \|f(x) - c_k\|^2$$

3

Pseudo-label Assignment

Use cluster ID as pseudo-label

$$y_{\text{pseudo}} = \text{cluster_id}$$

4

Network Update

Train CNN with pseudo-labels (classification)

$$L = -\log P(y_{\text{pseudo}} | x)$$



Problem: Trivial Solutions

All samples collapse into a single cluster



Solution: Empty Cluster Reassignment

Fill empty clusters with samples from the largest cluster



SwAV Operating Principle

1. Multi-Crop Augmentation

 224×224 (×2) 96×96 (×4)

Generate multiple views with various crop sizes

2. Prototype Assignment

$$\mathbf{q} = \text{softmax}(\mathbf{z} \cdot \mathbf{C} / \tau)$$

\mathbf{z} : feature, \mathbf{C} : prototypes, τ : temperature

Soft-assign each view to prototypes

3. Swapped Prediction

$\mathbf{z}_1 \rightarrow \text{predict } \mathbf{q}_2$

$\mathbf{z}_2 \rightarrow \text{predict } \mathbf{q}_1$

Predict View 2 from View 1's code (and vice versa)

4. Loss Computation

$$\mathcal{L} = \ell(\mathbf{z}_1, \mathbf{q}_2) + \ell(\mathbf{z}_2, \mathbf{q}_1)$$

Cross-entropy between predictions



Online Clustering

Update prototypes every batch (no K-means needed)



Sinkhorn-Knopp

Ensures balanced cluster assignment (equipartition)



Memory Efficiency

No need to store negative pairs (vs SimCLR)