Supplementary Information: How and What to Learn: Taxonomizing Self-Supervised Learning for 3D Action Recognition

1. Momentum Contrastive SSL

Here, we revisit the contrastive learning model that we used in more details. Contrastive learning generally relies on the number of negative samples to generate robust representations. Different strategies have been recently proposed to collect a set of negative samples against a positive pair during training [1, 2]. Of particular interest is the Momentum Contrast (MoCo) [2] which enables the learning of large and consistent dictionaries showing great promises for contrastive image representation learning. The approach is based on a momentum encoder that builds a dictionary as a queue of encoded keys such that the current mini-batch gets enqueued while the oldest mini-batch dequeued. The dictionary keys are selected on-the-fly by a set of latent samples in the batch during training. The parameters of the momentum encoder are updated during training as: $\theta_k \leftarrow m\theta_k + (1-m)\theta_q$, where $m \in [0, 1)$ is the momentum coefficient controlling the magnitude of update at each iteration and θ_q and θ_k are the parameters of the query and key encoders, respectively. Here, only θ_q is updated by back-propagation and the momentum update ensures that θ_k evolves more smoothly than θ_q, which ensures consistency of key representations over different iterations.

Given input sample x, a contrastive objective function has low value when the query $\mathcal{E}(x)$ is similar to its positive key $\mathcal{K}_+$ (i.e., *attraction*) and dissimilar to all other negative keys (i.e., *repulsion*). We follow [2] and measure similarities by dot products $\mathcal{S}(X, Y) = exp(X \cdot Y/\tau)$ and consider the InfoNCE contrastive loss function [3] in our work:

$$\begin{aligned}\mathcal{L}_C &= -log\left(\frac{exp(\mathcal{E}(x)\cdot\mathcal{K}_+/\tau)}{\sum_{i=0}^{N}\exp(\mathcal{E}(x)\cdot\mathcal{K}_i/\tau)}\right) \\ &= -\mathcal{E}(x)\cdot\mathcal{K}_+/\tau + log(\sum\exp(\mathcal{E}(x)\cdot\mathcal{K}_i/\tau)),\end{aligned} \tag{1}$$

where τ is a temperature coefficient and N is the dictionary size. The sum is over one positive and $N-1$ negative samples. For simplicity, we can omit the parameter τ and consider only attraction and repulsion terms in Eq. 1. Hence contrastive learning can be rewritten as:

$$\mathcal{L}_C = \underbrace{-\mathcal{S}(\mathcal{E}(x), \mathcal{K}_+)}_{attraction} + \underbrace{log(\sum\exp(\mathcal{S}(\mathcal{E}(x), \mathcal{K}_-)))}_{repulsion} \tag{2}$$

2. Model Training

Algorithm 1: Pseudocode of the proposed model.

```
# x: input minibatch x
# x1, x2: transformed instances of x
# Enc_q, Enc_k: Query and key encoders.
# Dec: decoder network
# Q: dictionary as a queue of K keys (dxK)
# m: momentum
# w_1, w_2: regularization coefficients
# t: temperature
for (x, x_1, x_2) in data_loader:
    # Encoding
    z_gt = Enc_q(x) # ground truth (Nxd)
    z_q = Enc_q(x1) # queries (Nxd)
    z_k = Enc_k(x2).detach() # keys (Nxd)

    # Decoding
    x_rec = Dec(z_gt) # input reconstruction

    # positive logits: Nx1
    l_pos = einsum([q, k])
    # negative logits: NxK
    l_neg = einsum([q, Q.detach()])

    # logits: Nx(1+K)
    logits = cat([l_pos, l_neg], dim=1)

    # Compute loss
    labels = zeros(N)
    # Contrastive Bottom-Up objective
    L_CB = cross_entropy(logits/t, labels)
    # Attractive Bottom-Up objective
    L_AB = 2-2*(z_q*z_gt).sum()
    # Attractive Top-Down objective
    L_AT = ((x_rec-x)**2).sum()/N

    Loss = L_CB+w_1*L_AB+w_2*L_AT

    # update Enc_q and Dec networks
    Loss.backward()
    update(Enc_q.params)
    update(Dec.params)

    # update Enc_k
    Enc_k.params = m*Enc_k.params+(1-m)*Enc_q.params

    # update dictionary
    enqueue(Q, z_k) # enqueue the current minibatch
    dequeue(Q) # dequeue the earliest minibatch
```

Our best performing model is a combination of Attractive $\mathcal{E}$, Contrastive $\mathcal{E}$, and Attractive $\mathcal{D}$ objective functions. For

$\mathcal{L}_{AE}$	$\mathcal{L}_{CE}$	$\mathcal{L}_{AD}$	$\mathcal{L}_{CD}$	**UCLA**	**NTU (1-layer)**	**NTU (3-layer)**	**NTU (1024 units)**
✓				59.98	34.63	12.35	36.08
✓		✓	✓	60.37	50.70	22.01	51.78
✓			✓	76.81	37.93	10.07	43.31
			✓	77.73	42.84	38.88	48.31
	✓		✓	79.56	66.93	55.66	66.88
✓	✓		✓	79.96	67.0	56.33	63.88
✓		✓		80.15	59.31	61.51	62.67
✓	✓			80.74	67.03	59.06	68.61
	✓	✓	✓	82.43	59.87	48.86	59.95
✓	✓	✓	✓	82.98	63.09	55.85	61.40
		✓		83.45	59.19	61.90	62.58
	✓			83.73	66.54	55.37	67.21
		✓	✓	85.50	48.22	20.08	48.90
	✓	✓		85.70	66.88	56.61	66.59
✓	✓	✓		**86.08**	**67.06**	59.99	**68.64**

Table 1. The complete taxonomy of self-supervised learning objective functions for action recognition. Here we also include models (in red) which combine both Attractive and Contrastive objective functions in the same space (of either the Encoder or the Decoder). First and second columns are experiments performed using our proposed model on UCLA and NTU, respectively. The third and fourth columns are experiments performed on NTU with deeper (3 layers) and wider (1024 units) models, respectively.

more clarity, we provide a detailed pseudo-code in Algorithm 1.

3. Additional experiments

Taxonomy: Taxonomy experiments presented in the paper (Table 1 and 6) consist of combinations of four objective functions (Attractive versus Contrastive, in $\mathcal{E}$ or $\mathcal{D}$). In each of $\mathcal{E}$ and $\mathcal{D}$, we chose to discuss models with either Contrastive or Attractive objective functions since the Contrastive one contains already an attractive term (see Eq.2). Here, for more insights we present the full set of $\sum_{k=1}^{K} \frac{4!}{k!(4-k)!} = 15$ combinations including the remaining models which are highlighted in red in Table 1. We summarized all results in the same table whose columns correspond to the following experiments (with a left to right order): experiments with our proposed model on UCLA dataset, experiments with the same model on NTU dataset, experiments on NTU with a deeper model (3 layers), and finally NTU experiments on a wider model with 1024 units. From these results, we can see that there is no remarkable improvements when using both Attractive and Contrastive objectives within the same space (of $\mathcal{E}$ and $\mathcal{D}$).

Taxonomy analysis: Fitting a GLM to the taxonomy results allows us to evaluate the relationship between different SSL approaches and model performance. Of the taxonomy components, only the "Contrastive Decoder" ($t = 2.394, p < 0.038$, 1-tailed test) was significantly correlated with performance, although including this component yield worse overall performance than the other components. Ultimately, the best performing model on UCLA combines separate "Contrastive and Attractive Encoder" objective functions, with an "Attractive Decoder" objective function.

Attractive encoder loss collapsing: Tables 1 and 6 in the paper and Table 1 in SI show that the *Attractive Encoder* model achieves the lowest performance. One possible interpretation could be the fact that representations collapse to 0 and satisfies the loss. To further explore this hypothesis, we performed experiments on models whose weights are randomly initialized or trained on the $\mathcal{L}_{AE}$ objective while varying model depth. We notice an accuracy drop for all model depths (Table 2) which is consistent with this hypothesis.

# layers	**Random Init**	$\mathcal{L}_{AE}$
1	52.01	51.89
2	56.51	32.04
3	60.17	25.33
4	63.70	28.45
5	60.11	28.39

Table 2. Classification accuracy on UCLA of randomly initialized models vs. *Attractive Encoder* models with varying depths.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. Feb. 2020.

[2] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[3] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. July 2018.