# SUPPLEMENTARY MATERIAL

## A. Pseudocode

**Algorithm 1:** DDCL Pseudocode, PyTorch-like

```
# f:  encoder
# g:  projection mlp
# q:  prediction mlp
# _I, _V: DIR, DVR
# DR: disentangling ratio
# off_diagonal:  off-diagonal elements

for x in loader:
    # two randomly augmented versions of x
    x1, x2 = aug(x), aug(x)
    y1, y2 = f(x1), f(x2) # representations

    d_I = y1.size(1)*DR # DIR dimension

    # grouping and disentangling
    y1_I, y2_I = y1[:,:d_I], y2[:,:d_I] # DIR
    y1_V, y2_V = y1[:,d_I:], y2[:,d_I:] # DVR

    # projections for DIR and DVR
    z1_I, z2_I = g_I(y1_I), g_I(y2_I)# NxD
    z1_V, z2_V = g_V(y1_V), g_V(y2_V)# NxD

    # ------if method is Barlow Twins------
    # Batch Normalization
    z1_I, z2_I = BN(Z1_I), BN(z2_I)
    z1_V, z2_V = BN(Z1_V), BN(z2_V)

    # Cross-Corr matrix
    C = mm(z1_I.T, z2_I)/N # DxD
    C_diff = (C - eye(C.size(0))).pow(2)
    off_diagonal(C_diff).mul_(lambda)

    L_BT = C_diff.sum() # loss of BT
    DDL_sym = D(z1_V, z2_V)
    L_sym = gamma*DDL_sym + L_BT(z1_I, z2_I)

    # --------if method is Simsiams--------
    # predictions
    p1_I, p2_I = q_I(z1_I), q_I(z2_I)
    p1_V, p2_V = q_V(z1_V), q_I(z2_V)

    # Stop-grad
    z1_I, z2_I = z1_I.detach(), z2_I.detach()
    z1_V, z2_V = z1_V.detach(), z2_V.detach()

    S1 = -cos_simiarity(p1_I, p2_I)
    S2 = -cos_simiarity(p2_I, p1_I)

    L_Sims = S1/2 + S2/2 # loss of Sims
    DDL_asy = D(p1_V,z2_V)/2+D(p2_V,z1_V)/2
    L_asy = gamma*DDL_asy + L_Sims

def D(r1, r2):
    return abs(cos_similarity(r1, r2) - eps)
```

## B. Robustness in STL-10

| Method \ Distortion | CJ | CJ + Flip | CJ + 90° | CJ + 180° |
|---|---|---|---|---|
| Trained by BAug | | | | |
| BT | 90.62 | 90.56 | **43.29** | **37.10** |
| DDCL_Sym_DIR | 90.43 | 90.47 | 41.07 | 34.87 |
| DDCL_Sym | 90.74 | 90.64 | 41.70 | 35.61 |
| Simsiam | 90.84 | 90.94 | 42.34 | 36.56 |
| DDCL_Asy_DIR | 91.06 | 91.12 | 42.25 | 36.15 |
| DDCL_Asy | **91.19** | **91.22** | 42.36 | 36.72 |
| Trained by CAug | | | | |
| BT | 87.76 | 87.80 | 85.23 | 80.79 |
| DDCL_Sym_DIR | 87.57 | 87.32 | 85.55 | 80.90 |
| DDCL_Sym | 88.29 | 88.32 | 85.75 | 80.82 |
| Simsiam | 88.52 | 88.37 | 86.63 | 82.18 |
| DDCL_Asy_DIR | 89.26 | 89.38 | 87.47 | 82.28 |
| DDCL_Asy | **89.57** | **89.48** | **87.55** | **82.69** |

Table B.1. Robustness evaluation by different augmentation strategies on STL-10. 90° and 180° denote randomly applying -90° to 90° and -180° to 180° rotation during inference, respectively.

## C. Ablation Study on Warm-up

| Warm-up | CIFAR-10 | | CIFAR-100 | | STL-10 | |
|---|---|---|---|---|---|---|
| | w/o | w/30 | w/o | w/30 | w/o | w/30 |
| Simsiam | 91.83 | 91.56 | 65.99 | 66.29 | 91.22 | 91.02 |
| DIR_only | 91.75 | 92.01 | 65.77 | 65.66 | 91.26 | 91.28 |
| DDCL_Asy | 91.84 | **92.19** | 66.20 | **66.49** | **91.43** | 91.39 |

Table C.1. Effect of warm-up on the linear evaluation.

**Warm-up** We use a warm-up of 30 epochs in the default design to accompany the cosine training schedule. As shown in Tab. C.1 and Fig. C.1 , the overall performance of DIR_only and DDCL_Asy is generally improved by the warm-up strategy, and outperforms that of Simsiam.

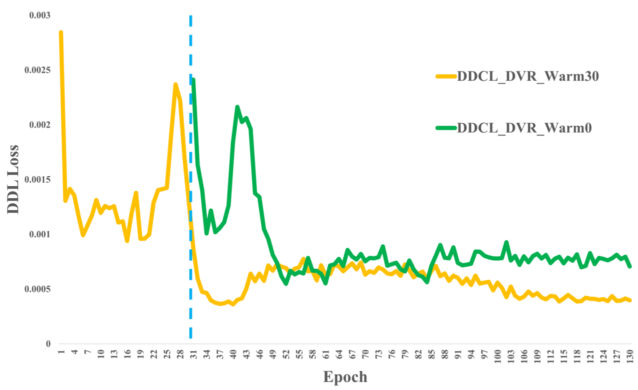


Figure C.1. The influence of Warm-up on Distortion-Disentangled Loss in DDCL_Asy when pre-training.

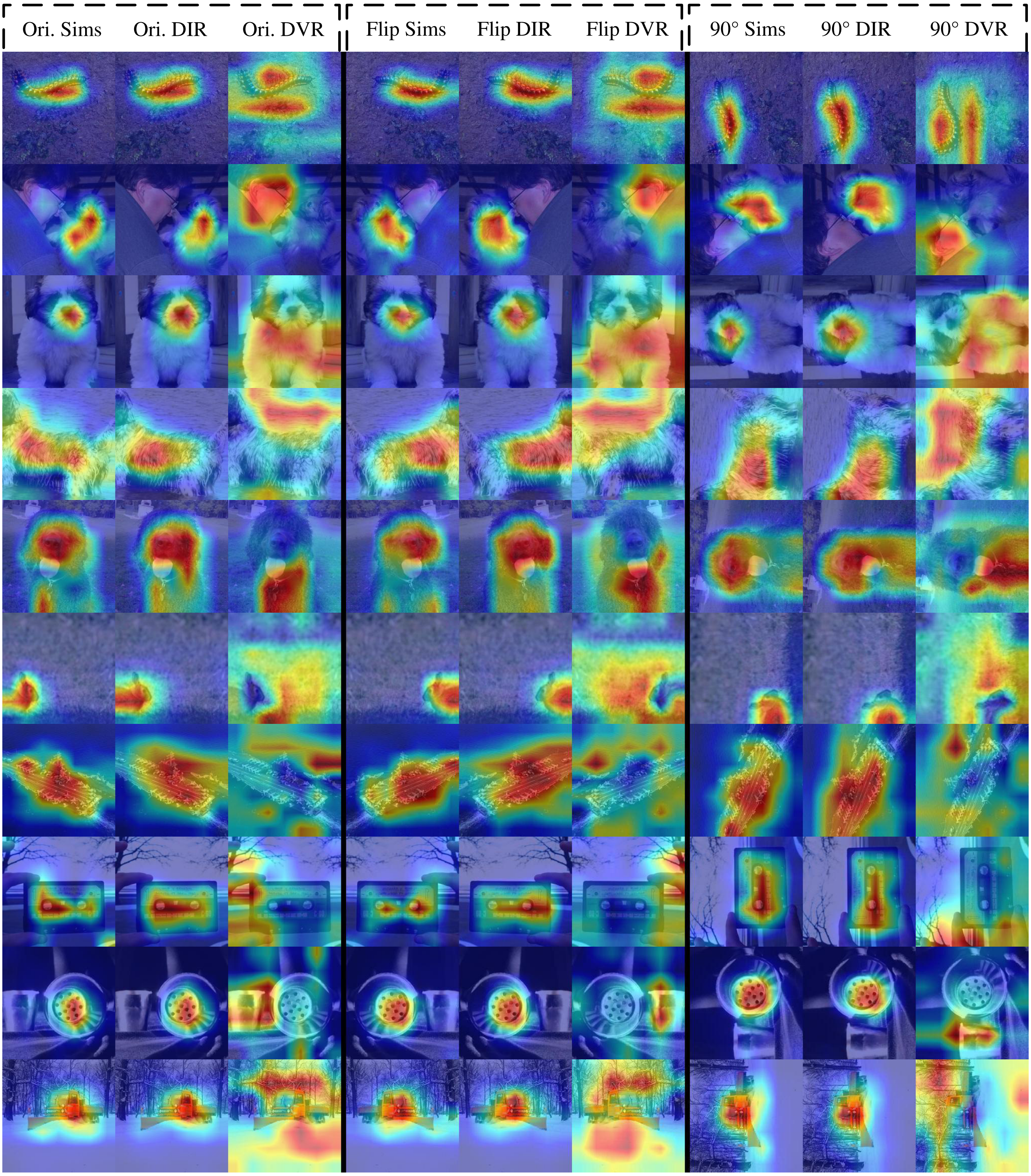


Figure D.1. In order to analyze the semantics of DVR, we report more attention maps of Simsiam and DIR and DVR of DDCL Asy of inputs with different distortions on ImageNet (pre-trained using CAug). 'Ori.', 'Flip' and '90°' represent that the input is original, horizontally flipped, and rotated by 90°.

## D. Semantic Analysis of DVR

We analysis more attention maps to study the semantics referred to by DVR as shown in Figure D.1. The semantics of DVR's regions of interest can be considered: the environment background, some components of the object, other foregrounds, and the "envelope" of the target object.

It is easy to understand that information about the environmental background and some components of the target object can provide additional information to improve the performance of the prediction (perhaps involving fairness analysis). The information of other foregrounds may give the model potential for multi-object detection and robustness to label noise (needs further exploration). In addition, we argue that this kind of "envelope" wrapping the target object can bring certain distortion information to the model (*i.e*., the rotation and flipping distortion in the figure).