

1188  
1189**SUPPLEMENTARY MATERIAL**

1190

**A. Pseudocode**

1191

---

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

---

```

1194    # f: encoder
1195    # g: projection mlp
1196    # q: prediction mlp
1197    # -I, -V: DIR, DVR
1198    # DR: disentangling ratio
1199    # off_diagonal: off-diagonal elements
1200
1201    for x in loader:
1202        # two randomly augmented versions of x
1203        x1, x2 = aug(x), aug(x)
1204        y1, y2 = f(x1), f(x2) # representations
1205
1206        d_I = y1.size(1)*DR # DIR dimension
1207
1208        # grouping and disentangling
1209        y1_I, y2_I = y1[:, :d_I], y2[:, :d_I] # DIR
1210        y1_V, y2_V = y1[:, d_I:], y2[:, d_I:] # DVR
1211
1212        # projections for DIR and DVR
1213        z1_I, z2_I = g_I(y1_I), g_I(y2_I) # NxD
1214        z1_V, z2_V = g_V(y1_V), g_V(y2_V) # NxD
1215
1216        # -----if method is Barlow Twins-----
1217        # Batch Normalization
1218        z1_I, z2_I = BN(z1_I), BN(z2_I)
1219        z1_V, z2_V = BN(z1_V), BN(z2_V)
1220
1221        # Cross-Corr matrix
1222        C = mm(z1_I.T, z2_I)/N # DxN
1223        C_diff = (C - eye(C.size(0))).pow(2)
1224        off_diagonal(C_diff).mul_(lambda)
1225
1226        L_BT = C_diff.sum() # loss of BT
1227        DDL_sym = D(z1_V, z2_V)
1228        L_sym = gamma*DDL_sym + L_BT(z1_I, z2_I)
1229
1230        # -----if method is Simsiam-----
1231        # predictions
1232        p1_I, p2_I = q_I(z1_I), q_I(z2_I)
1233        p1_V, p2_V = q_V(z1_V), q_V(z2_V)
1234
1235        # Stop_grad
1236        z1_I, z2_I = z1_I.detach(), z2_I.detach()
1237        z1_V, z2_V = z1_V.detach(), z2_V.detach()
1238
1239        S1 = -cos_simiarity(p1_I, p2_I)
1240        S2 = -cos_simiarity(p2_I, p1_I)
1241
1242        L_Sims = S1/2 + S2/2 # loss of Sims
1243        DDL_asy = D(p1_V, z2_V)/2+D(p2_V, z1_V)/2
1244        L_asy = gamma*DDL_asy + L_Sims
1245
1246    def D(r1, r2):
1247        return abs(cos_simiarity(r1, r2) - eps)

```

---

1238

1239

1240

1241

**B. Robustness in STL-10**

1242

1243

1244

1245

1246

1247

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

Method \ Distortion	CJ	CJ + Flip	CJ + 90°	CJ + 180°
Trained by BAug				
BT	90.62	90.56	<b>43.29</b>	<b>37.10</b>
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	<b>91.06</b>	<b>91.12</b>	42.25	36.15
DDCL_Asy	<b>91.19</b>	<b>91.22</b>	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	<b>87.47</b>	<b>82.28</b>
DDCL_Asy	<b>89.57</b>	<b>89.48</b>	<b>87.55</b>	<b>82.69</b>

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	<b>92.19</b>	66.20	<b>66.49</b>	<b>91.43</b>	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.

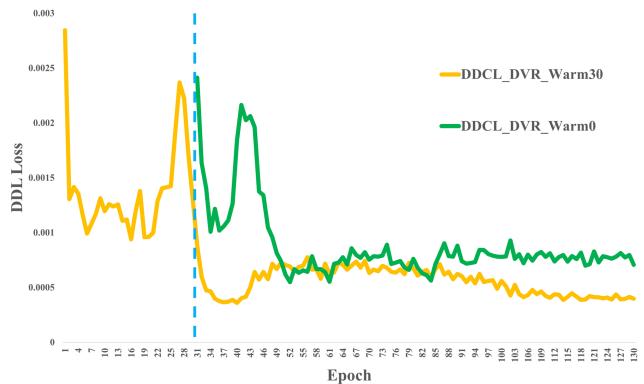


Figure C.1. The influence of Warm-up on Distortion-Disentangled Loss in DDCL\_Asy when pre-training.

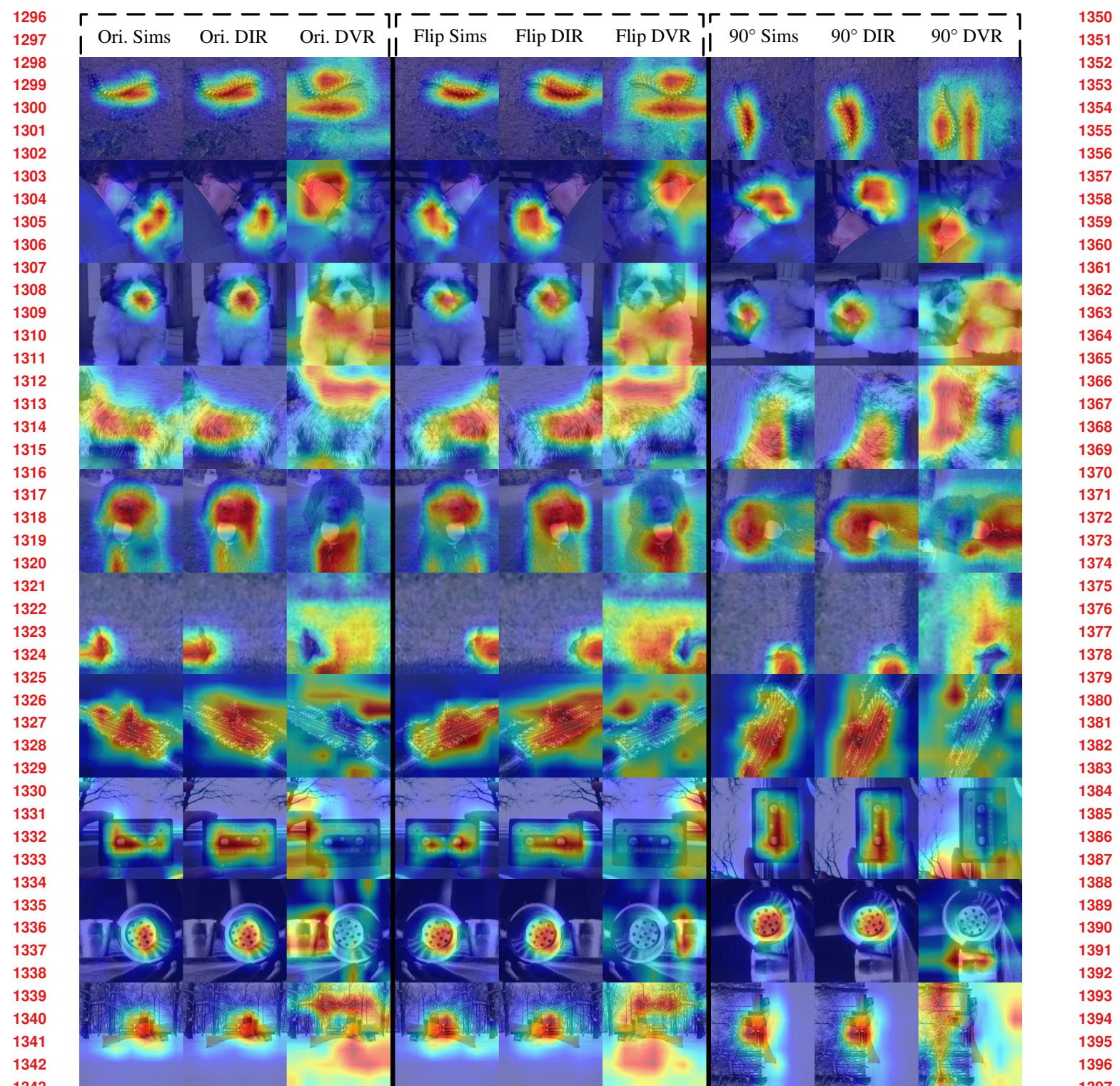


Figure D.1. In order to analyze the semantics of DVR, we report more attention maps of Simsiam and DIR and DVR of DDCL Asy of inputs with different distortions on ImageNet (pre-trained using CAug). ‘Ori.’, ‘Flip’ and ‘90°’ represent that the input is original, horizontally flipped, and rotated by 90°.

1348

1349

1404  
1405

## D. Semantic Analysis of DVR

1406  
1407  
1408  
1409  
1410

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.

1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420

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).

1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
14571458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511