A Algorithm

Algorithm 1: PyTorch-style pseudocode for DeCUR.

```
# f1, f2: encoder networks
# BN, N, K: batch normalization, batch size and embedding dimension
# on_diag, off_diag: on- and off-diagonal elements of a matrix

# loss function for common and intra-modal
def loss_c(C, lambda):
    l_on = (on_diag(C)-1).pow(2).sum()
    l_off = off_diag(C).pow(2).sum()
    return l_on + lambda x l_off

# loss function for unique
def loss_u(C, lambda):
    l_on = on_diag(C).pow(2).sum()
    l_off = off_diag(C).pow(2).sum()
    return l_on + lambda x l_off

# training
for x1,x2 in loader: # load a batch pairs
    (x1_1, x1_2), (x2_1, x2_2) = augment1(x1), augment2(x2)
    # compute embeddings and normalize
    z1_1, z1_2 = BN(f1(x1_1)), BN(f1(x1_2))
    z2_1, z2_2 = BN(f2(x2_1)), BN(f2(x2_2))
    # cross-correlation matrices
    C1 = z1_1.T @ z1_2 / N # KxK
    C2 = z2_1.T @ z2_2 / N # KxK
    Cm = z1_1.T @ z2_1 / N # KxK
    Cc = Cm[:K_c,:K_c] # KcxKc
    Cu = Cm[K_c:,K_c:] # KuxKu
    # calculate losses
    L1 = loss_c(C1,lmb1) # intra-modal M1
    L2 = loss_c(C2,lmb2) # intra-modal M2
    Lc = loss_c(Cc,lmbc) # cross-m. common
    Lu = loss_u(Cu,lmbu) # cross-m. unique
    loss = L1 + L2 + Lc + Lu # total loss
    # optimization
    loss.backward()
    optimizer.step()
```

B Implementation Details

B.1 Pretraining

SAR-optical pretraining SSL4EO-S12 [10] dataset is used for SAR-optical pretraining: Sentinel-1 GRD (2 bands VV and VH) and Sentinel-2 L1C (13 multispectral bands). The pixel resolution is united to 10 meters. We compress and normalize the optical data to 8-bit by band-wise mean and standard deviation; for SAR data, we cut out 2% outliers for each image and normalize it by band-wise mean and standard deviation.

Standard ResNet50 is used as the encoder backbone, of which the first layer is modified to fit the input channel number. The projector is a 3-layer MLP, of which the first two layers include Linear, BactchNorm and ReLU, and the last one includes only a linear layer. We adopt two residual deformable attention (RDA) modules after the last two blocks of the ResNet encoder. Following [13], for the first RDA module, we use a feature map 14×14, 8 heads with 128 channels each, 4 groups, stride 2, and kernel size 5; for the second RDA module, we use a feature map 7×7, 16 heads with 128 channels each, 8 groups, stride 1, and kernel size 3.

We use the LARS [15] optimizer with weight decay 1e-6 and momentum 0.9. We use a learning rate of 0.2 for the weights and 0.0048 for the biases and batch normalization parameters. We reduce the learning rate using a cosine decay schedule [7] (no warm-up periods). The biases and batch normalization parameters are excluded from LARS adaptation and weight decay.

RGB-DEM pretraining The training split of the GeoNRW [1] dataset is used for RGB-DEM pretraining: aerial orthophoto (3 bands RGB) and lidar-derived digital elevation model (1 band height). The pixel resolution is 1 meter. We use standard ResNet50 without modifying the input layer (i.e., we duplicate the DEM image to 3 channels). Other model architecture and optimization protocols are the same as SAR-optical pretraining.

RGB-depth pretraining SUN-RGBD [8] dataset is used for RGB-depth pretraining: indoor RGB and depth images. Following [16], we preprocess the depth images to HHA format [4]. We use standard ResNet50 and MiT-B2/B5 from SegFormer as the backbones. For segformer backbones, we use AdamW optimizer and a learning rate of 1e-4.

Data augmentations We follow common augmentations in the self-supervision literature [3] for optical and RGB images (resized crop, color jitter, grayscale, Gaussian blur, horizontal and vertical flip, color drop, solarize), and remove infeasible ones for special modalities. Specifically, for SAR, we use random resized crop, grayscale, Gaussian blur, and horizontal and vertical flip; for DEM images, we use random resized crop and horizontal and vertical flip; for HHA images, we use random resized crop and horizontal flip.

B.2 Transfer learning

SAR-optical transfer learning We evaluate SAR-optical pretraining on the BigEarthNet-MM [9] dataset for the multi-label scene classification task.

We compress and normalize the optical images to 8-bit by band-wise mean and standard deviation; for SAR images, we cut out 2% outliers for each image and normalize it by band-wise mean and standard deviation. As the optical data of BigEarthNet-MM is Sentinel-2 L2A product (12 bands), we insert one empty band to match the pretrained weights (13 bands). We use common data augmentations including RandomResizedCrop (scale 0.8 to 1) and RandomHorizontalFlip.

Standard ResNet50 is used as the encoder backbone for each modality, of which the first layer is modified to fit the input channel number, and the last layer is modified as an identity layer. The encoded features are concatenated, followed by a fully connected layer outputting the class logits. The encoders are initialized from the pretrained weights. For linear classification, the encoder weights are frozen and only the last classification layer is trainable; for fine-tuning, all weights are trained.

We optimize MultiLabelSoftMarginLoss with batch size 256 for 100 epochs. We use the SGD optimizer with a weight decay of 0 and momentum of 0.9. The learning rate is 0.5 for linear classification, and 0.05 for fine-tuning. We reduce the learning rate by factor 10 at 60 and 80 epochs.

RGB-DEM transfer learning We evaluate RGB-DEM pretraining on the GeoNRW dataset for the semantic segmentation task. We use common data augmentations including RandomResizedCrop (scale 0.2 to 1) and RandomHorizontalFlip.

Fully convolutional networks (FCN) [6] with standard ResNet50 backbone for each modality are used as the segmentation model. The last three feature maps from both modalities are concatenated and upsampled to the input size. They are further followed by 1x1 convolution outputting three segmentation maps, which are added together to form the final output map. The encoders are initialized from the pretrained weights. For linear classification, the encoder weights are frozen; for fine-tuning, all weights are trainable.

We optimize CrossEntropyLoss with batch size 256 for 30 epochs. We use the AdamW optimizer with a weight decay of 0.01. The learning rate is 0.0001 for both linear classification and fine-tuning.

RGB-depth transfer learning We evaluate RGB-depth pretraining on SUN-RGBD and NYU-Depth v2 datasets for the semantic segmentation task. We use common data augmentations including RandomResizedCrop and RandomHorizontalFlip.

FCN with ResNet50 backbones is used as the segmentation model for single-modal RGB semantic segmentation. We optimize CrossEntropyLoss with batch size 8 for 40k iterations. We use the SGD optimizer with weight decay 1e-5. The learning rate is 0.01 with polynomial decay for fine-tuning.

CMX [16] with segformer [14] backbones are used as the segmentation model for RGBD semantic segmentation. We follow the same settings of CMX for SUN-RGBD and NYU-depth v2 datasets.

C Additional results

Additional results with ViT backbones To complement experiments with ResNet backbones in the main paper, we provide in Tab. 1 additional frozen-encoder results with ViT backbones on SAR-optical and RGB-DEM scenarios. The consistent improvement of DeCUR over SimCLR verifies again its architecture-agnostic property.

Table 1: Additional frozen-encoder results with ViT backbones.

	Backbone	BigEarthNet-SAR-1%	GeoNRW-RGB-1%
Rand. Init.	ViT-S/16	57.7	13.2
SimCLR	ViT-S/16	73.9	35.1
DeCUR (ours)	ViT-S/16	**75.8**	**35.6**

Additional results with attention mechanisms To further support the benefits of adopting deformable attention, we provide in Tab. 2 additional frozen-encoder results with different attention designs. Specifically, our design is better than the popular convolutional block attention module (CBAM) [11]. In addition, we also explore the possibility of cross-modal attention: taking queries from one modality, and keys and values from another modality, which results in performance decreasing. While this cross-attention tends to be beneficial in supervised learning [2,5], The reason could be a potential information leak that hurts the self-supervised pretraining task.

Table 2: Additional frozen-encoder results with different attention designs. *DA* means deformable attention without residual connection; *Cross DA* represents cross-modal deformation attention; *RDA* represents deformable attention with residual connection.

	BigEarthNet-SAR-1%	GeoNRW-RGB-1%
DA [12]	73.5	30.2
Cross DA	72.5	29.6
CBAM [11]	73.1	31.0
RDA (ours)	**74.4**	**32.2**

D Explainability analysis

D.1 Algorithm

For a better understanding of our explainability implementation, we provide united pseudocode of 1) cross-modal representation alignment, 2) t-SNE representation visualization, 3) spatial and spectral saliency statistics based on Grad-Cam and Integrated Gradients, in Algorithm 2.

Algorithm 2: Pseudocode for DeCUR explainability.

```
# f1,f2: encoder networks
# BN: batch normalization
# N,K: batch size and embedding dimension
# on_diag: on-diagonal elements of a matrix
# IG: Integrated Gradients

# 1.Cross-modal representation alignment
def alignment_histogram(z1, z2):
   C = z1.T @ z2 / N # KxK
   losses = (on_diag(C)-1).pow(2) # Kx1
   return histogram(losses, range=(0,1))

# 2.Representation visualization
def tsne_vis(z1, z2):
   feature = torch.cat((z1,z2),-1) # Nx2K
   feature = feature.permute(1,0) # 2KxN
   return tsne(feature, n_components=2)

# 3.Spatial saliency visualization
def gradcam_vis(x1):
   z1 = BN(f1(x1)) # NxK
   z1_c = z1[:,:Kc].mean(dim=-1) # Nx1
   z1_u = z1[:,Kc:].mean(dim=-1) # Nx1
   out1 = torch.cat((z1_c,z1_u),-1) # Nx2
   gc1 = LayerGradCam(f1, f1.last_conv2)
   attr1_c = gc1.attribute(x1,target=0) # Nx7x7
   attr1_u = gc1.attribute(x1,target=1) # Nx7x7
   return upsamp(attr1_c), upsamp(attr1_u)
   # Nx224x224, Nx224x224

# 4.Spatial saliency statistics
def gradcam_stat(x1,x2):
   att1_c, att1_u = gradcam_vis(x1)
   att2_c, att2_u = gradcam_vis(x2)
   mul_c = norm(att1_c) x norm(att2_c) # Nx1
   mul_u = norm(att1_u) x norm(att2_u) # Nx1
   return mul_c, mul_u

# 5.Spectral saliency statistics
def IG_stat(x1):
   # define "IG_vis" similar to gradcam
   att1_c, att1_u = IG_vis(x1) # NxCx224x224
   imp_c = att1_c.mean(dim=(0,2,3)) # NxC
   imp_u = att1_u.mean(dim=(0,2,3)) # NxC
   return imp_c, imp_u
```

D.2 Additional examples

Below we show some additional explainability examples. Note that the decoupling and matching results depend on the samples. Specifically, some images have a strong overlap between modalities (potentially more common dimensions) while others tend to be more orthogonal (potentially more unique dimensions, decoupling helps more).

Cross-modal alignment histograms See Figure 1.

t-SNE representation visualization See Figure 2.

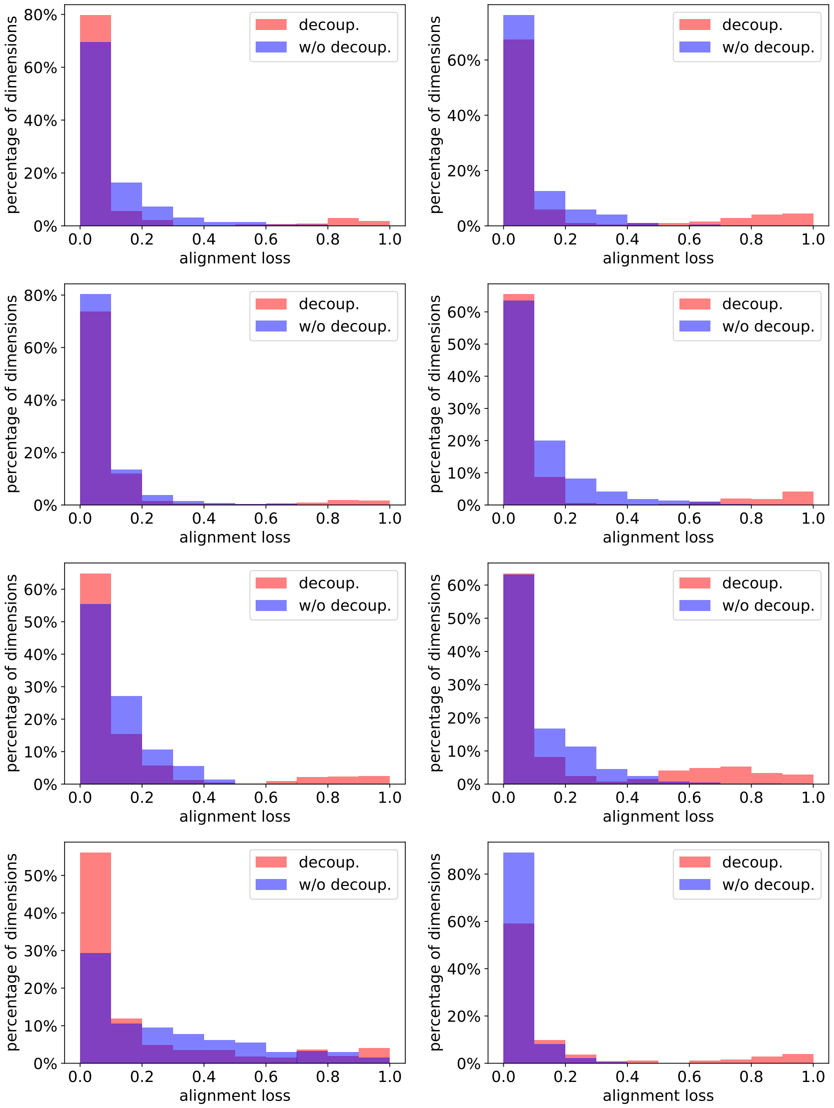

Fig. 1: Cross-modal representation alignment histograms of 4 batches of samples. Left: SAR-optical; right: RGB-DEM.

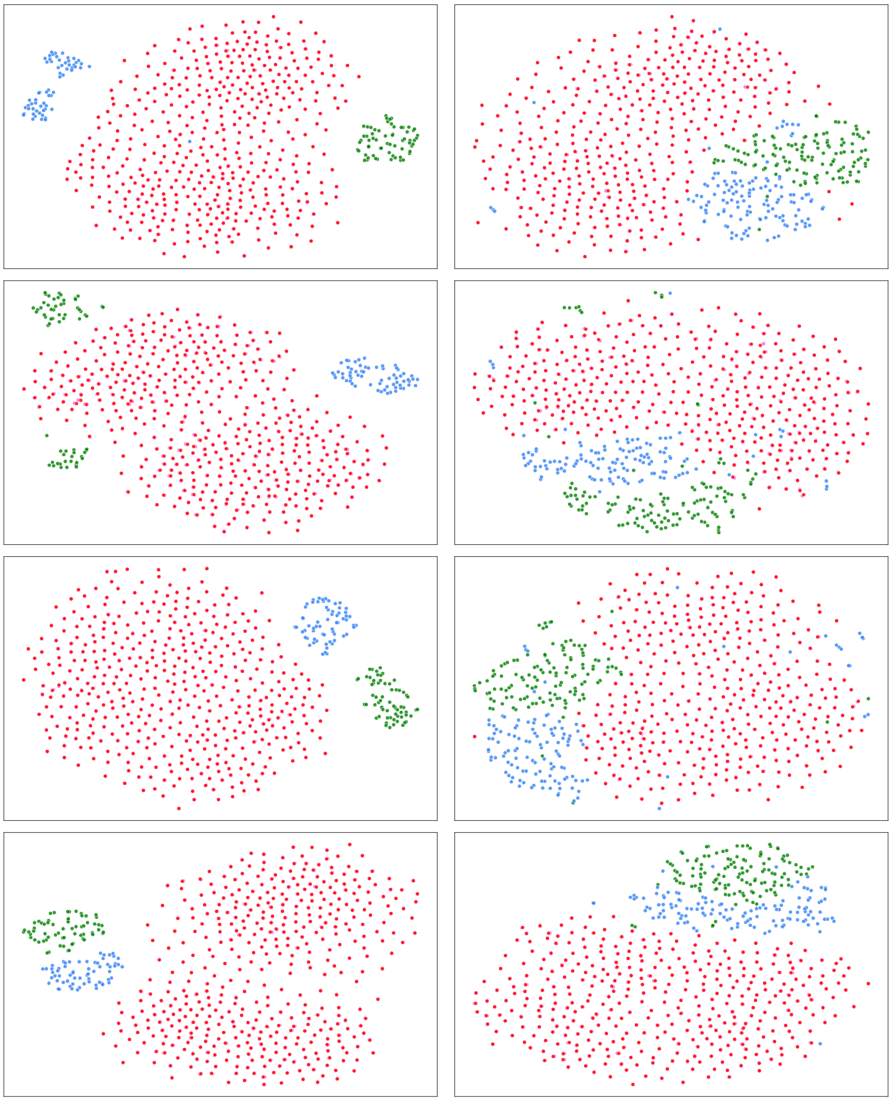

Fig. 2: t-SNE representation visualization of 4 batches of samples. Left: SAR-optical; right: RGB-DEM.

References

1. Baier, G., Deschemps, A., Schmitt, M., Yokoya, N.: Geonrw (2020). https://doi.org/10.21227/s5xq-b822, https://dx.doi.org/10.21227/s5xq-b822 2
2. Feng, Z., Song, L., Yang, S., Zhang, X., Jiao, L.: Cross-modal contrastive learning for remote sensing image classification. IEEE Transactions on Geoscience and Remote Sensing (2023) 4
3. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020) 2
4. Gupta, S., Girshick, R., Arbeláez, P., Malik, J.: Learning rich features from rgb-d images for object detection and segmentation. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13. pp. 345–360. Springer (2014) 2
5. Jha, A., Bose, S., Banerjee, B.: Gaf-net: improving the performance of remote sensing image fusion using novel global self and cross attention learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 6354–6363 (2023) 4
6. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015) 3
7. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016) 2
8. Song, S., Lichtenberg, S.P., Xiao, J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 567–576 (2015) 2
9. Sumbul, G., De Wall, A., Kreuziger, T., Marcelino, F., Costa, H., Benevides, P., Caetano, M., Demir, B., Markl, V.: Bigearthnet-mm: A large-scale, multimodal, multilabel benchmark archive for remote sensing image classification and retrieval [software and data sets]. IEEE Geoscience and Remote Sensing Magazine **9**(3), 174–180 (2021) 2
10. Wang, Y., Braham, N.A.A., Xiong, Z., Liu, C., Albrecht, C.M., Zhu, X.X.: SSL4EO-S12: A large-scale multi-modal, multi-temporal dataset for self-supervised learning in earth observation. arXiv preprint arXiv:2211.07044 (2022) 1
11. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018) 4
12. Xia, Z., Pan, X., Song, S., Li, L.E., Huang, G.: Vision transformer with deformable attention. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4794–4803 (2022) 4
13. Xia, Z., Pan, X., Song, S., Li, L.E., Huang, G.: Dat++: Spatially dynamic vision transformer with deformable attention. arXiv preprint arXiv:2309.01430 (2023) 2
14. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. Advances in Neural Information Processing Systems **34**, 12077–12090 (2021) 3
15. You, Y., Gitman, I., Ginsburg, B.: Large batch training of convolutional networks. arXiv preprint arXiv:1708.03888 (2017) 2
16. Zhang, J., Liu, H., Yang, K., Hu, X., Liu, R., Stiefelhagen, R.: Cmx: Cross-modal fusion for rgb-x semantic segmentation with transformers. arXiv preprint arXiv:2203.04838 (2022) 2, 3