

Lab 3

实验要求

- 1.基于DANN分别进行域适应和域泛化的迁移学习，在SEED数据集上做跨被试留一交叉验证。
- 2.移除DANN中的域判别器/域分类器，重新训练模型，在SEED数据集上做跨被试留一交叉验证，与1中的实验结果对比。
- 3.选择交叉验证的某一折，分别可视化1和2中DANN特征提取器提取出的数据特征（可用 tsne 将数据特征降至二维以可视化），观察不同被试数据的分布变化。

实验结果

模型构建

```
class DANNModel(nn.Module):
    def __init__(self, input_size, hidden_size1, hidden_size2, num_classes):
        super(DANNModel, self).__init__()
        self.feature_extractor = nn.Sequential(
            nn.Linear(input_size, hidden_size1),
            nn.ReLU(),
        )
        self.class_classifier = nn.Sequential(
            nn.Linear(hidden_size1, hidden_size2),
            nn.ReLU(),
            nn.Linear(hidden_size2, num_classes)
        )

        self.domain_classifier = nn.Sequential(
            nn.Linear(hidden_size1, hidden_size2),
            nn.ReLU(),
            nn.Linear(hidden_size2, 1)
        )

    def forward(self, x, alpha):
        features = self.feature_extractor(x)
        class_logits = self.class_classifier(features)
        domain_logits = self.domain_classifier(features * alpha)

        return class_logits, domain_logits
```

在域适应和域泛化的DANN模型中，我们均采用简单的MLP模型做特征提取器，其由一个线性层和ReLU层构成，标签分类器和域分类器的结构均为线性层加ReLU层加线性层组成。具体参数如下：

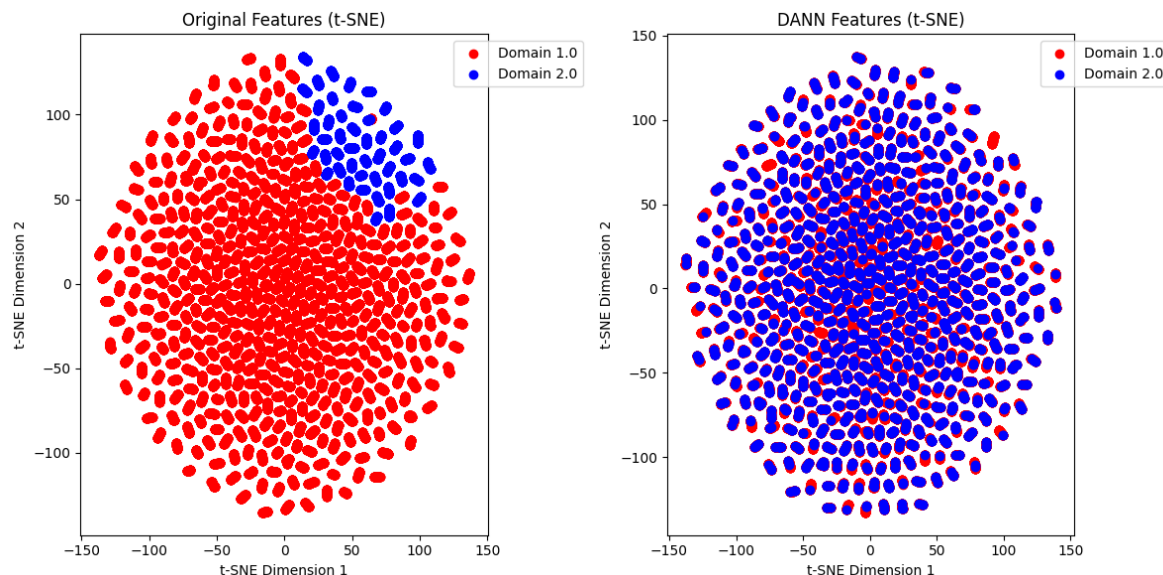
```
input_size = 310 #输入大小
hidden_size1 = 100 #隐藏层1大小
hidden_size2 = 50 #隐藏层2大小
num_classes = 3 # 分类数
num_epochs = 10 #迭代次数
lr = 0.001 #学习率
```

域适应

有域分类器

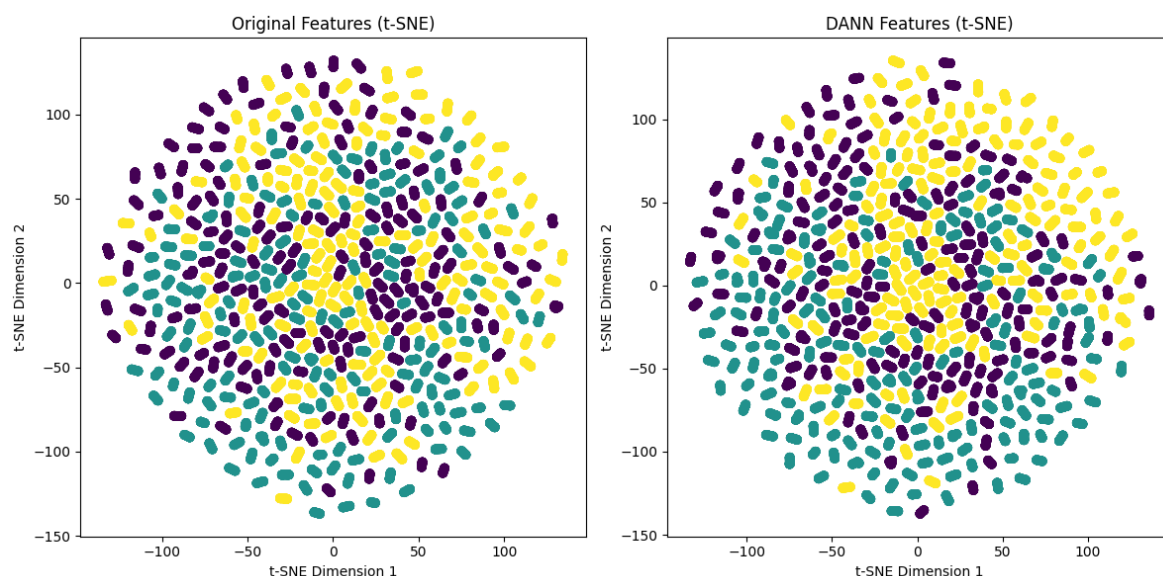
我们将整个测试集看作目标域，11个训练集看作一个源域。除0外（此时等于没有域分类器），在领域自适应参数alpha取0.4时域适应的DANN效果最好。

域标签可视化



我们可以看到原始的源域和目标域界限分明，而在采用了域适应的DANN后，源域和目标域较为均匀的混合在了一起。

Label可视化



我们可以看到和原始的相比，在采用了域适应的DANN后，三类label聚拢了一些，边界更分明。

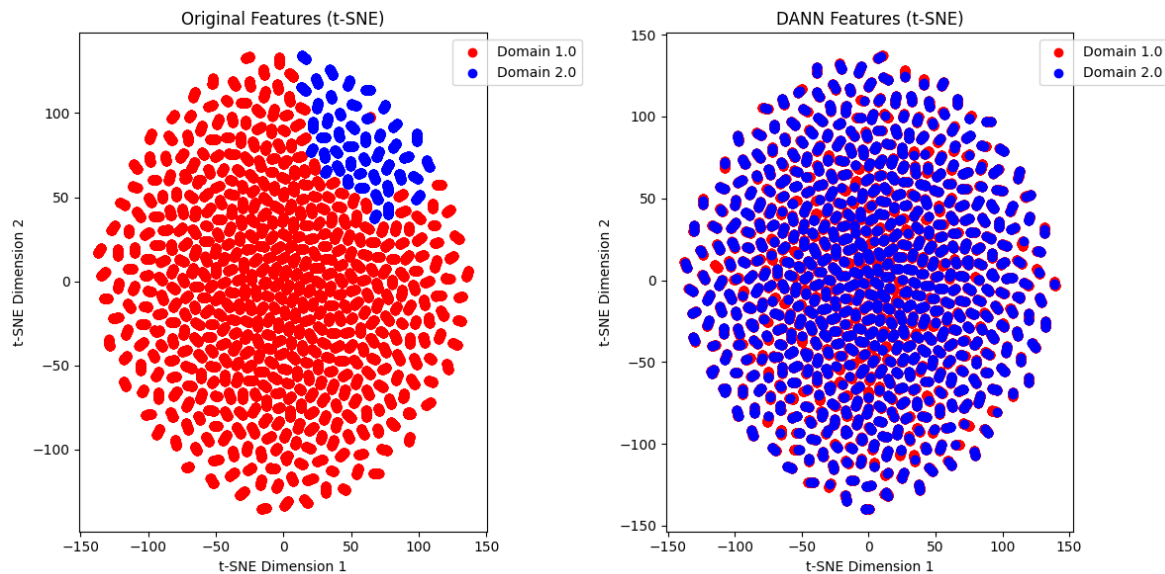
准确率和标准差

```
PS E:\desktop\lab3> python -u "e:\desktop\lab3\DA.py"
Testing Subjects: 1004
平均准确率: 0.528955[3280549
标准差: 0.13261023949026451
```

使用简单的MLP作为特征提取器的情况下，采用域适应的DANN准确率达到了53.0%，比直接猜三分类高了很多，学习效果较为明显，是一个相对合理的结果，标准差也在合理范围内。

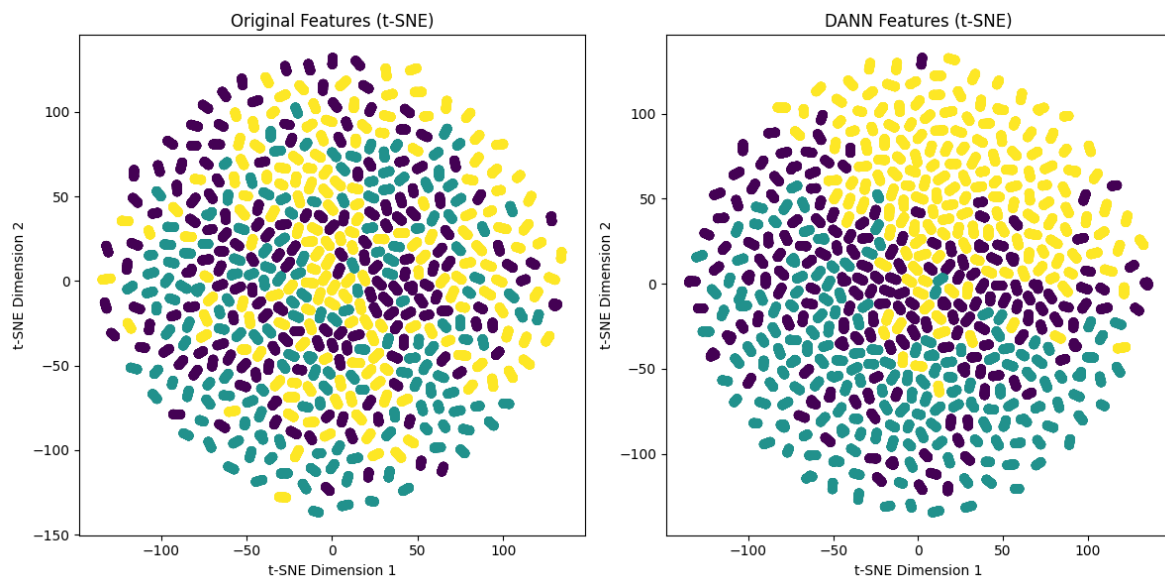
无域分类器

域标签可视化



我们可以看到原始的源域和目标域界限分明，而使用了删除域分类器的域适应的DANN后，源域和目标域还是较为均匀的混合在了一起，和不删除域分类器的效果差不多。

Label可视化



我们可以看到在删除了域分类器后，三类label聚拢效果更好了，边界更分明。

准确率和标准差

```
PS E:\desktop\lab3> python -u "e:\desktop\lab3\DA without domain classifier.py"
Testing Subjects: 100%
平均准确率: 0.555720906730088
标准差: 0.12976435789851098
```

我们可以看到在删除了域分类器后，准确率达到了55.6%，比不删除的还要高，标准差也降低了。

结果讨论

我们在这里发现删除了域分类器后，得到的效果反而更好，删除了域分类器就是不在进行域适应，相当于直接使用简单的MLP做特征提取和只用标签分类器。域适应反而出现了负优化，我们通过域标签的可视化发现其实目标域和源域相差可能相差不大，不采用域分类器的组别也能很好的混合均匀，域适应此时增强模型适应性的效果就不会明显。而域适应因为损失函数里添加了域分类损失的项，可能会导致标签的损失影响降低，对于源域的学习效果下降，又因为源域和目标域本身就较为接近，所以在目标域上

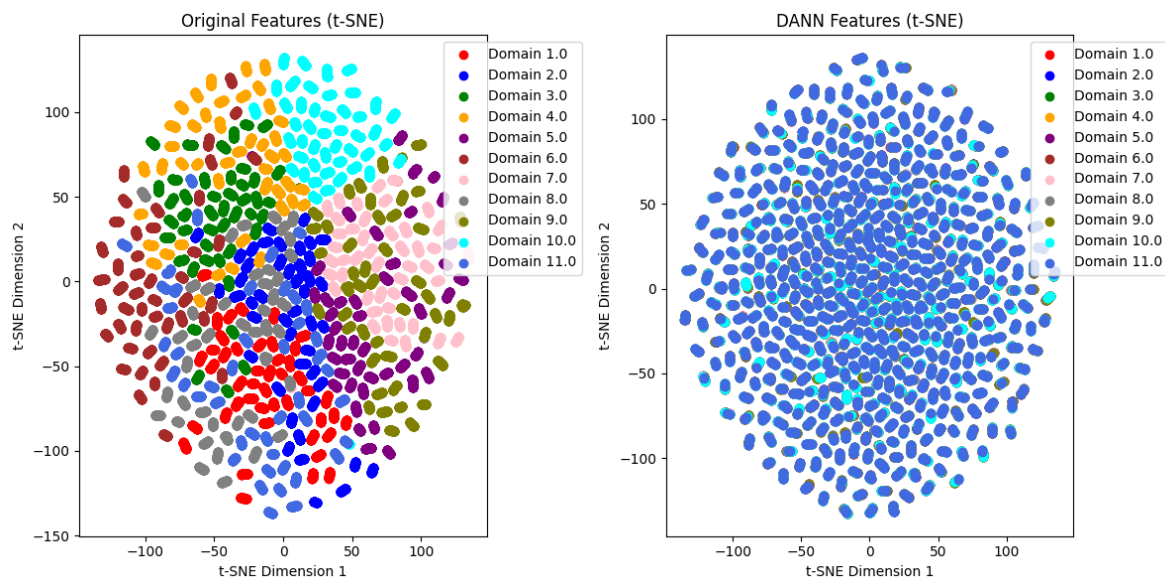
准确率也会下降。综上所述，因为源域和目标域本身混合较好的缘故，增强适应性的效果不明显，又影响了对源域的学习，所以删除域分类器后效果反而更好。

域泛化

有域分类器

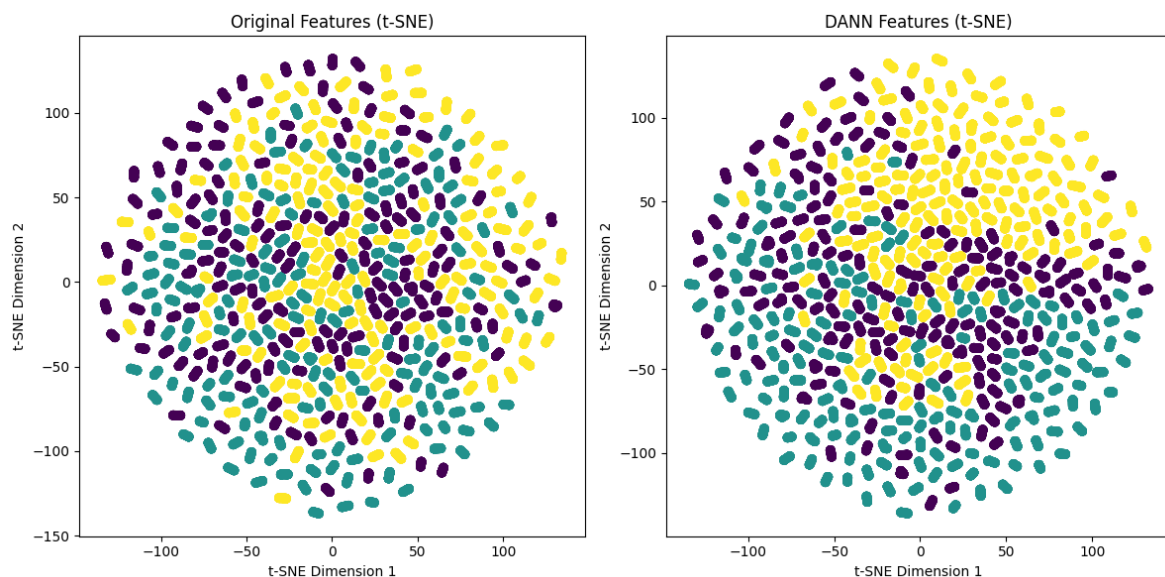
我们将11个训练集看作11个源域,在领域自适应参数alpha取0.395时域泛化的DANN效果最好。

域标签可视化



我们可以看到原始的11个源域间界限分明，而在采用了域泛化的DANN后，11个源域都均匀的混合在了一起，我们可以明显看到不同源域之间的覆盖与重叠，域泛化效果很好。

Label可视化



我们可以看到和原始的相比，在采用了域泛化的DANN后，三类label聚拢效果较为明显，边界也明显了不少。

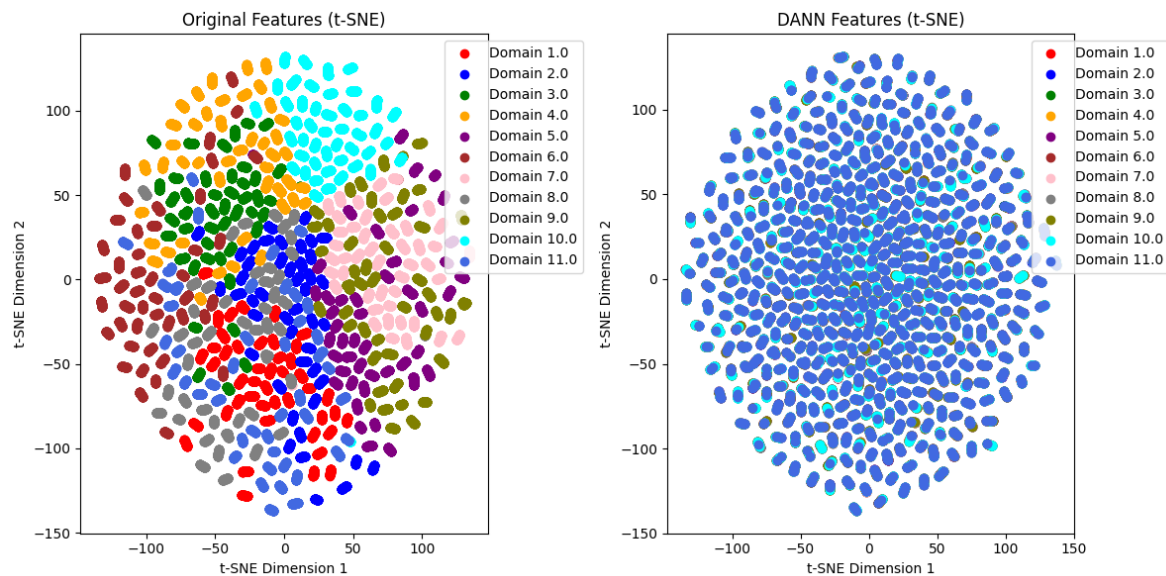
准确率和标准差

```
PS E:\desktop\lab3> python -u "e:\desktop\lab3\06.py"
Testing Subjects: 100%
平均准确率: 0.565386643529164
标准差: 0.13175146675964594
12/12 [00:27<00:00, 2.32s/it]
```

使用简单的MLP作为特征提取器的情况下，采用域泛化的DANN准确率达到了56.5%，比直接猜三分类高了很多，也高于域适应的结果，学习效果较为明显，是一个相对合理的结果，标准差也在合理范围内，比域适应更小。

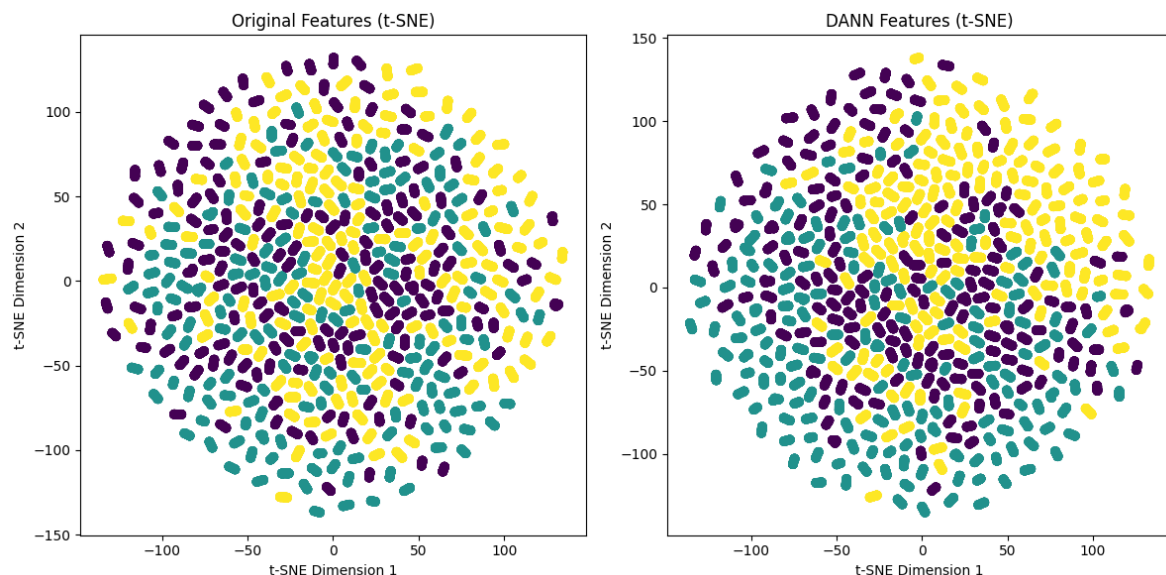
无域分类器

域标签可视化



我们可以看到在删除了域分类器后，11个源域依旧混合在一起，和之前相比，重合程度稍微下降了一点。

Label可视化



我们可以看到在删除了域分类器后，三类label，和之前相比更为分散，混合程度更高，边界不那么清晰。

准确率和标准差

```
PS E:\desktop\lab3> python -u "e:\desktop\lab3\DG without domain classifier.py"
Testing Subjects: 100%
平均准确率: 0.5445038268672474
标准差: 0.1443618112527856
```

我们可以看到在删除了域分类器后，准确率为54.4%，低于删除之前的，标准差也高于删除之前的，总体效果变差了，这与可视化显示出的结果也相匹配。

结果讨论

采用了域泛化的DANN效果是最好的，比删除了域分类器的效果好，比域适应效果也好。我们看删除域分类器后的域标签可视化发现，11个源域均匀混合效果本来就很好，说明源域之间的特征相差比较小，使得域泛化的领域迁移效果的学习就很好，又因为目标域和源域也相似，不会因为领域之间的差异大，导致领域迁移困难，在模型的泛化能力范围内。域泛化不会使用无标签的目标域数据，对源域学习是影响也不大，调节合适的参数会使得源域学习效果好，模型也有一定的泛化能力，这样就比删除了域分类器和域适应的效果要好。