**比赛的思路上，都是这样做的**

**整体思路上，我们是先用machine id，machine type来作为监督信号，使用AdaCos loss或者 Arcface loss来训练模型；随后把这个模型当成特征提取器；在embedding基础上，训练GMM模型；最后在GMM上得到异常分数**

**后续改进的点：**

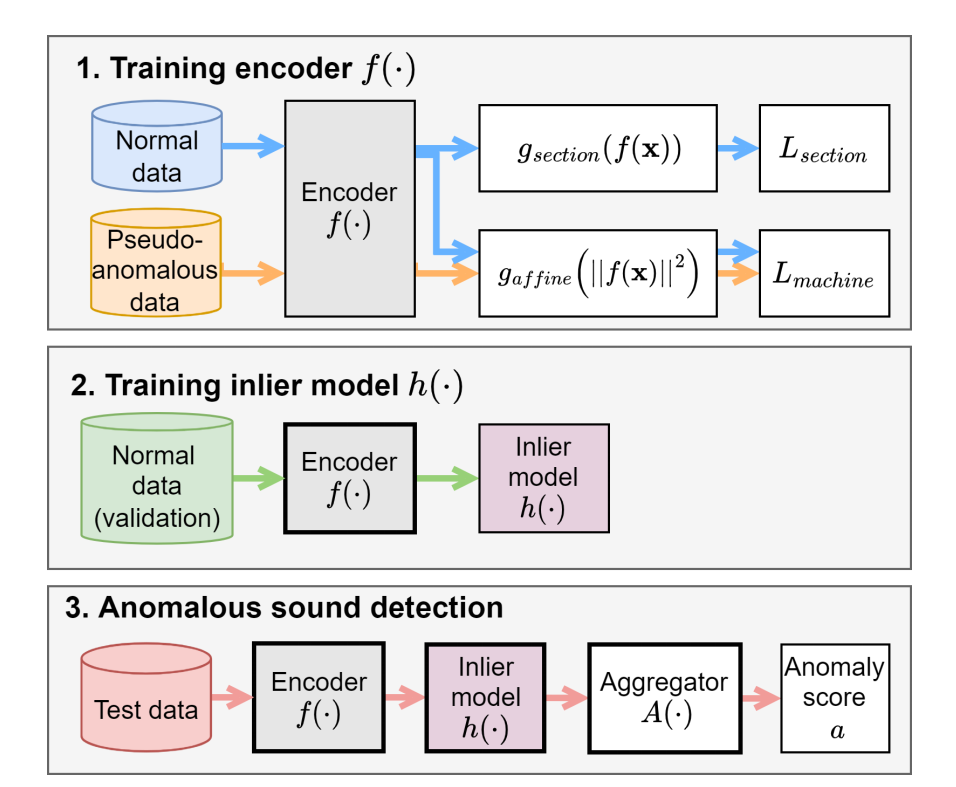
1. **数据端增加CMVN ————CODE DONE——Train-TEST Pass**
2. **特征构建端增加分类的类别，尽量把能用到的监督信号都用到 ———CODE DONE——Train-TEST Pass**
3. **特征构建端使用 Adacos loss来进行训练，避免超参数的选择——CODE DONE——Train-TEST Pass**
4. **检测端增加GMM——CODE DONE——Train-TEST Pass**
5. **想办法将logits和embedding想办法融合到一起————DONE**
6. **MixtureUp training（主要参考Sub-cluster Adacos这个论文）**

目前将1234的内容叠加到一起的话，相比于复现的比赛冠军方案，我们在auc 和 p-auc上，有

* Arcfaceloss有助于提升embedding之间的区分程度
* sample-wise temporal mean normalization similar to cepstral mean normalization (CMN)有助于提升模型对噪声的鲁棒性
* 训练GMM的时候，为了避免domain transfer，可以使用target-source样本的embedding mixture的方法来做数据增强

下图的这个思路就和我们剔除无关信息的思路是一样 的

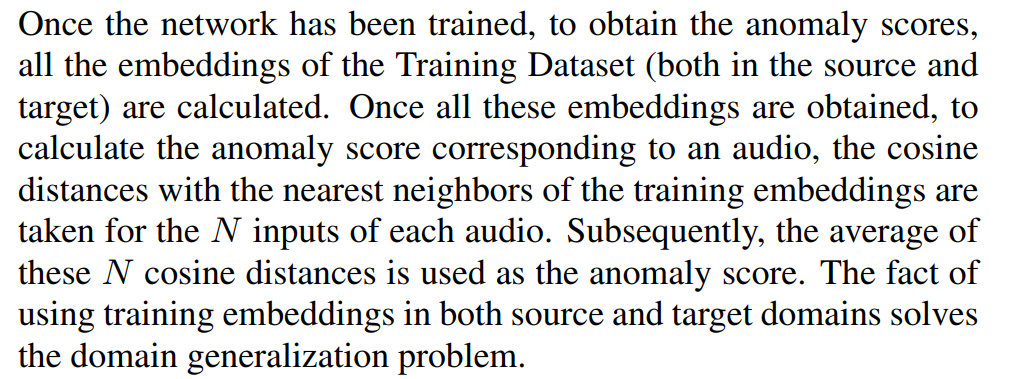
https://github.com/4uiiurz1/pytorch-adacos/blob/master/metrics.py



异常检测的部分【也就是Abnormal Score 的部分】

**异常赋分方法**

1. （VIT based）the cosine distance between this embedding and the nearest neighbor of the training audios is used as the anomaly score. 使用待测样本与最近训练样本邻居之间的embedding的cosine distance作为异常分数



1. Embedding后面接一个GMM模型，