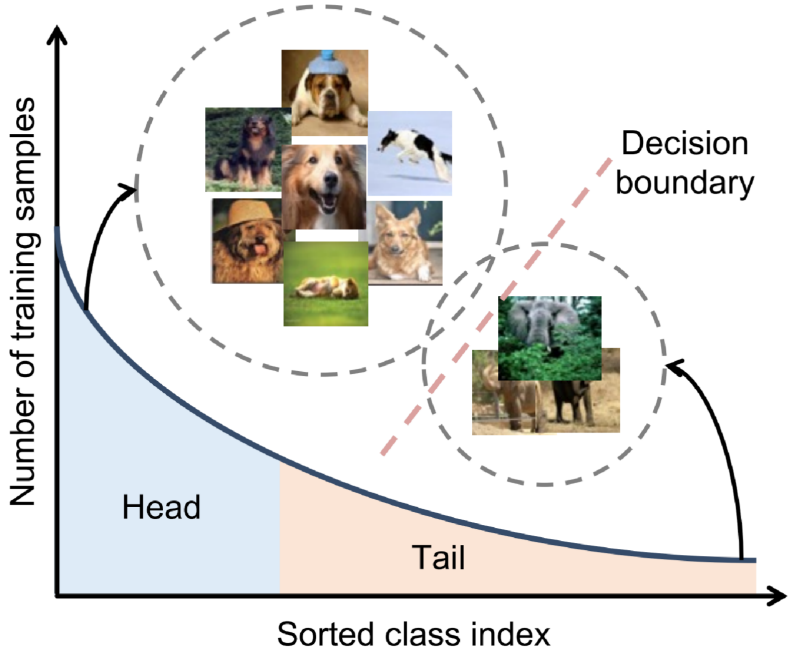
Deep Long-Tailed Learning-A Survey

（P.s. 笔记是根据自己理解写得，主要是写给自己看的所以话会比较多，如果有不对的地方请评论告诉我。）

（在最后总结的地方有放我自己整理的论文里的文章的分类流程图，流程图是用mermaid写得，大家可以按照自己需求调整，同时也有 github 上还在更新的 survey 链接）

Part 1：背景介绍



长尾问题其实就是数据不平衡的一种（如上图），长尾问题中的尾部类其实就是小样本。随着近年来深度学习模型在一些人造的平衡数据集如 ImgaeNet 上已经获得极高的准确率，所以越来越多的人在关注数据长尾分布这一更现实、更关乎落地的问题。长尾问题近几年发展的特别快，在最基本的问题上已经能得到很高的预测准确率，但是还是有未来的提升空间和一些未来新的深入研究方向。

关于常用的数据集在相关应用里有涉及，而常用的backbone在论文2.4节有提及。

Part 2：相关方法

1.论文中提到的几类经典方法已经整理好，具体的分类见笔记的Appendix。（每一个具体的方法都有一些标签，方便查阅）下面先简单提一下几个大类（里面的论文往往用了很多种trick，所以有一篇文章在多个类下面的情况）：

- 类别重均衡（涉及到的方法往往针对样本不均衡，希望改变一些值来直接消除这种不均衡带来的bias）：
    - 重采样（经典的一些采样策略，与模型相关的一些特殊的采样策略）
    - 改变损失函数（改变权重大小，对类别边缘分布做出限制）
    - 对分类过程中的logit做事后调整
- 信息增强：
    - 迁移学习（将头部信息迁移到尾部，做预训练，知识蒸馏，半监督之类的自训练）

    - 数据增强（这里的数据增强更侧重于是一些传统方法的数据增强，可以理解成对原始数据直接做数据增强。
- 模型提升：
    - 改进特征提取模型（度量学习，顺序学习，原型学习，迁移学习）
    - 改进分类模型
    - 改进两阶段解耦学习
    - 多个专家的集成学习

2.作者针对这些方法有一些简单的总结：

- 对于第一类的重均衡学习：
    - 优点：往往改动量不大比较容易，且方便部署嵌入别的模型使用；同时像一些改变loss的方法和改变logit的方法都是由严格的理论支撑的。
    - 缺点：并没有从本质上解决尾部类别仍是小样本这一问题，也即尾部和头部类别的信息量仍有很大差异，这也在一定程度上导致了我们往往顾此失彼（当然也与模型能力等有关），在提升尾部类别准确性时会牺牲头部类别准确率。
- 对于第二类信息增强：
    - 优点：从更加本质的方面，即数据信息量差异上，缓解了长尾问题，一定程度上避免了前面提到的顾此失彼的方法；同时，对于一些basic的传统数据增强方法，也是很容易部署嵌入别的模型使用。
    - 缺点：如果只是简单应用一些与类别无关的数据增强，那这样所有类别都能得到增强，反而会加剧类别不平衡；（另一方面，我觉得还有比如说像半监督学习这种，如果给的额外的未标签数据更加不平衡，那这样也是会加剧类别不均衡这一问题的。）
- 对于第三类模型提升：
    - 优点：解耦学习这个方法十分简单有效，把长尾问题分成特征提取和分类两个经典模块做处理；一些集成学习能实现同时提升尾部和头部准确率这一需求。
    - 缺点：对于解耦学习来说，两阶段这种模式不容易部署在其他现有任务中，并且有论文指出这种嵌入使用下，解耦学习并不是那么有效；对于集成学习来说，它的计算消耗会比较大（但是这个能通过一些trick缓解），另一方面我自己觉得集成学习还是太复杂，不够优美。

Part 3 : 进一步结合实验分析

1.论文里面新提出来了一个测试指标：**relative accuracy** $A_r = \frac{A_t}{A_u}$:

- A_t是原先的测试指标，即Top-1准确率。
- A_u是 upper reference accuracy ，也就是理论上最大的准确率，计算公式如下 $A_u = max(A_v, A_b)$:
    - A_v 是同一个模型框架结合交叉熵损失，在均匀分布的训练集（这里这个均匀分布数据集的样本数与原本的长尾训练数据集是一样的）上训练得到的训练精度。
    - A_b 是同一个模型，在和上面一样的均匀分布的训练集上训练得到的训练精度。
    - 理论上，在均匀分布数据集上获得的准确度相比于在长尾分布上，可以看作是是理论最高的准确度。
- 考虑到原先的 Top-1 准确率高的原因有可能是模型本身就很好，也有可能是这个数据集本身是十分均衡的，这样你就看不出这个模型是否对于处理长尾问题本身是有用的。所以说这个指标的目的是确定所提出的方法或者模型是不是真的对解决长尾问题有效。

- 举个例子来说，假如有两个模型，处理的是同一个数据集，A_u 都相等，这意味着这两个模型对于这个均匀的数据集来说性能或者说适配程度是一样的，所以这时候如果其中某一个的 A_t 比较高，那么就说明对应的这个方法更适合解决这个数据集的长尾问题。再举个极端的例子来说，如果有一个模型特别大，比如原本是 resnet-32 ，一下子给换成 resnet-152 ，那么如果它 Top-1 准确率变高，就说不清楚了。
- 提出这个指标和我感觉和我之前想法是一致的，我之前读一些长尾问题论文觉得，先前的长尾问题相关工作对于同一个数据集往往是基于同模型同参数来测试的，比如对于 CIFAR-LT 数据集一般用 resnet32（ epcoh 和 bacthsize 这些超参数也一般是一样的），但是像那些改进模型等等的，比如说集成学习这种，那拿原来的 TOP-1 准确率来对比感觉就不是很合适。
- 当然除以 A_u 只能说是缓解了上面提到的这个情况，我个人不确定会不会有过度修正的问题。

2.论文里面进行了一系列实验分析（在 ImageNet-LT 上进行实验的）：

- 首先单纯对比各种方法的性能，主要从原先的 TOP-1 准确率 A_t 、改进的相对准确率 A_r 两方面来看：
    - 在原先 TOP-1 准确率上，除了 Decouple-CB-CRT 和 BBN 以外的方法性能都高于 baseline 的 Softmax 。这里论文里解释说前者是因为 Decouple-CB-CRT 在第一阶段提取特征时是用类别均衡采样的方式来训练的，这解释得通，因为根据先前研究以及证明类别均衡这种方式不利于特征学习；对于后者，论文猜想是由于 BBN 最后整合两个 branch 的课程学习方式不合理，在最后的时候会过度focus在尾部类上（这点我保持怀疑，因为我看过 BBN 这篇文章，尽管 BBN 并未在 imagenet-lt 上实验，但它在消融实验的时候探讨过取不同的权重变化函数带来的影响，并且它的模型在 CIFAR-LT 和 iNaturalist 数据集上表现都不错。当然本篇论文作者在这里也是猜想，大家也可以提一下观点。）
    - 对于改进的相对准确率 A_r，实验结果首先验证里它提出的初衷，大部分方法的 upper reference accuracy (UA) 都差不多，但是SSP, MiSLAS, TADE 这几种方法的 UA 高一些，恰恰是因为它们用了数据增强或者说是模型提升。后面论文里就简单举了一个例子说明 UA 这个指标还是有参考意义的。这段最后，论文分析认为近年来各种长尾方法的 UA 总体上呈上升趋势，这说明长尾问题确确实实在不断被解决，先阶段对于长尾分布最有效的是集成学习 TADE ，但仍有提升空间。
- 论文接着对几类方法做了对比分析：
    - 对于第一类的重均衡方法，论文主要对 cost-sensitive 的方法分析了一下，总体上来说各种 cost-sensitive 方法的 UA 都差不多，所以用原本的 Top-1准确度就可以了。此外论文提了 Focal loss 对于类别数很多的这种长尾数据集表现不好，LDAM 方法中提出的 LDAM loss 要和论文里的 deferred scheme 一起使用才比较有效（因为我自己还没有仔细研读过这两篇 paper，只是大概了解它们的方法，所以这两点分析我也存疑，到时候如果读了这两篇论文有想法了再来写一点）。
    - 对于数据增强方法，像 SSP 这种迁移学习尽管带来了 UA 的提升，但是其相对准确率 A_r 提升非常大，这意味着这些方法的准确度提升还是很大程度上由于缓解了长尾问题。
    - 对于模型提升方法也是类似，集成学习 TADE 和 RIDE 在 Top-1 准确度和相对准确率 A_r 上都是目前的 SOTA。
- 和先前研究一样，本篇论文同样对head-class，middle-class，tail-class三个子数据集进行分析：
    - 大多数长尾问题方法还是以牺牲 head-class 准确度来提升 middle-class 和 tail-class 准确度；
    - 这当然不可取，从目前来看，信息增强和集成学习对于解决这个问题比较有效；
    - 目前的SOTA方法是集成学习 TADE，但是它并未在各个subset上都实现了 SOTA，作者认为这也许说明了要获得更好的长尾问题的模型性能，需要在所有类别之间寻求一个均衡。
    - 我自己认为不管是 head-class 还是 middel-class 和 tail-class 都还有一定的性能提升空间，最理想的模型应该是能在这三个subset上都实现 SOTA 。

Part 4 : 相关应用

论文在这里主要提的是解决长尾分布问题的方法在视觉相关的应用（当然在 NLP 等领域也是有很多的应用，句子里面的单词这种数据本身就是长尾的）：

- 图像分类：
  - 多分类数据集：Imagenet , cifar , places , iNaturalist
  - 应用：表情识别，医疗图像诊断...
  - 多标签数据集：VOC-LT ， COCO-LT
  - 应用：网络图像分类，人脸属性分类...
- 物体识别和分割：
  - 数据集：LVIS , COCO
  - 应用：点云分割，城市场景理解，无人机检测
- 视觉关系学习：
  - 应用：场景图生成， visual question answering ， image captioning

Part 5 : 未来方向

论文在这里主要介绍了关于长尾分布的一些未来方向，一方面是方法上还能有哪些提升，另一方面是一些与长尾分布相关的新问题：

- 首先是方法的未来考虑方向上：
  - class re-balancing: 这方面需要尽量避免使用 label frequencies 作为先验知识， 因为对于一些复杂的情景，如多标签分类或者说物体识别上，并不能事先直接获得和多分类任务中类似的 label frequencies。
  - Transfer learning: 考虑利用更多的无标签数据来做迁移学习是个可行的方向。论文同样提到，现有的一些该类方法在多标签分类或者说物体识别中不一定能实现。
  - data augmentation: 对于数据增强同样是需要找到一个泛化性强的方法，以应对各类长尾任务。
  - ensemble learning: 集成学习有机会实现在所有类别上获得精度提升，值得更深入研究。
- 接着是长尾分布相关的一些任务展望：
  - 第一个是如何应对未知分布的测试集，在这里测试集的分布不仅未知而且是任意的，也就是说训练集和测试集分布可能存在较大偏差。在这个问题上，论文提到 LADE 是可以应对任意分布的测试集，但需要事先已知其分布；TADE 是可以处理完全未知的任意的测试集（这里因为我还没看这两篇paper，不知道 TADE 是否已经不错的解决了这个issue，后面会看 TADE ， 到时候补上看法）。
  - 第二个是开放长尾识别，OLTR 这篇文章有提到，也就是和开集识别相关，比如说需要考虑一些未见到的类别，这样就得确保模型不会把信息较少的尾部类和这些新类混淆。
  - 第三个是与联邦问题相关，主要涉及不同用户的本地数据分布是不同的（联邦学习我不是特别了解）。
  - 第四个是指在现实应用中，很多类别的数据不是一下子就可以得到的，而是按照一定顺序陆陆续续获得的。这我理解是一个动态获取信息的过程，一方面这样会导致数据分布无法事先获取，另一方面是在学习后面的类别时会忘记先前的类别信息。
  - 第五个是指我们获取的数据可能并不是从一个数据分布中获得的，而是属于多个不同的数据分布中。
  - 第六个是指样本数据肯定带着噪音，这就意味着模型的鲁棒性会变得非常差，尤其是对于尾部数据。
  - 第七个指长尾回归问题，即标签是连续的。

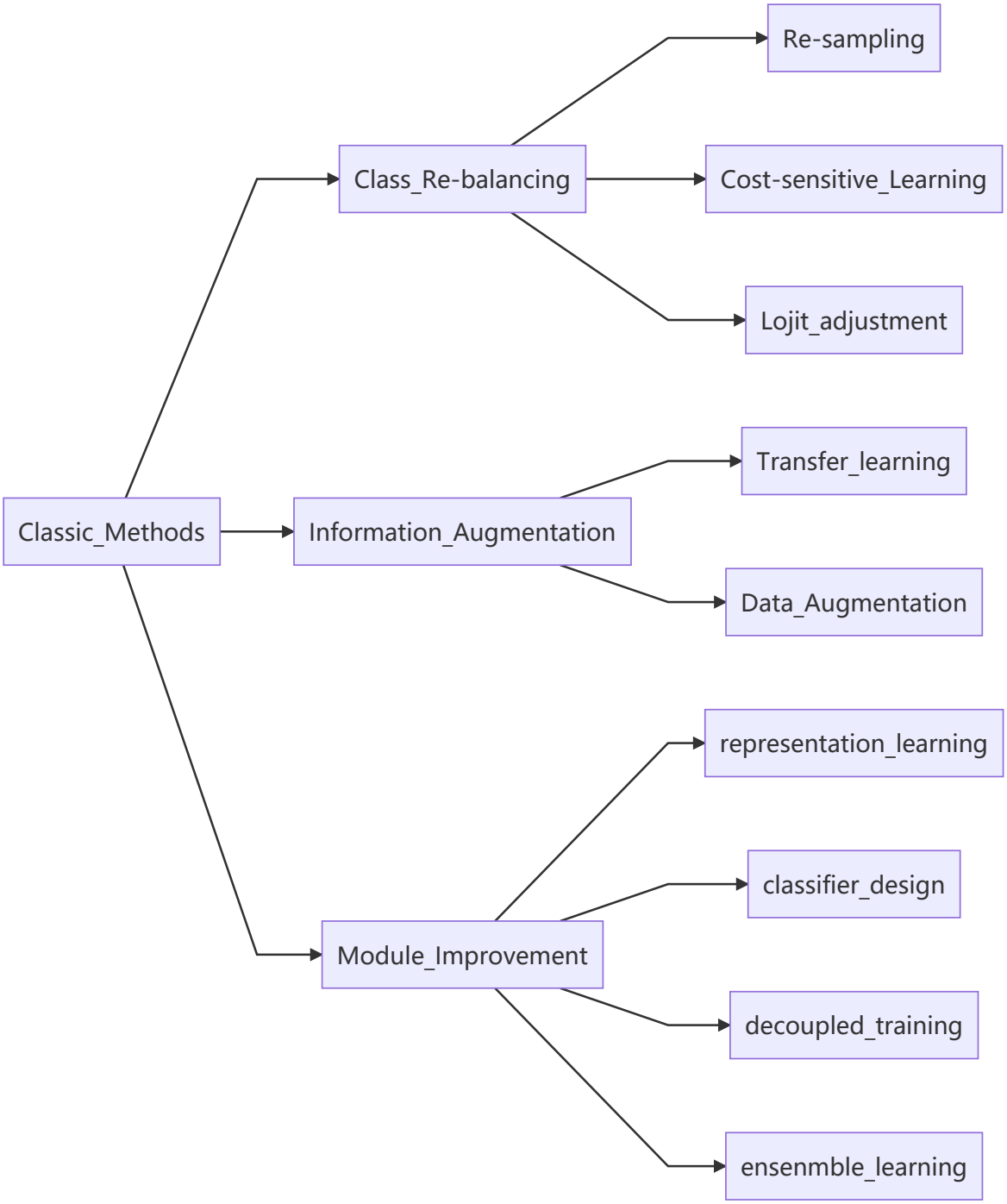
- 第八个是长尾视频学习，近期已有长尾视频数据集 VideoLT 被提出。

Part 6 : 总结

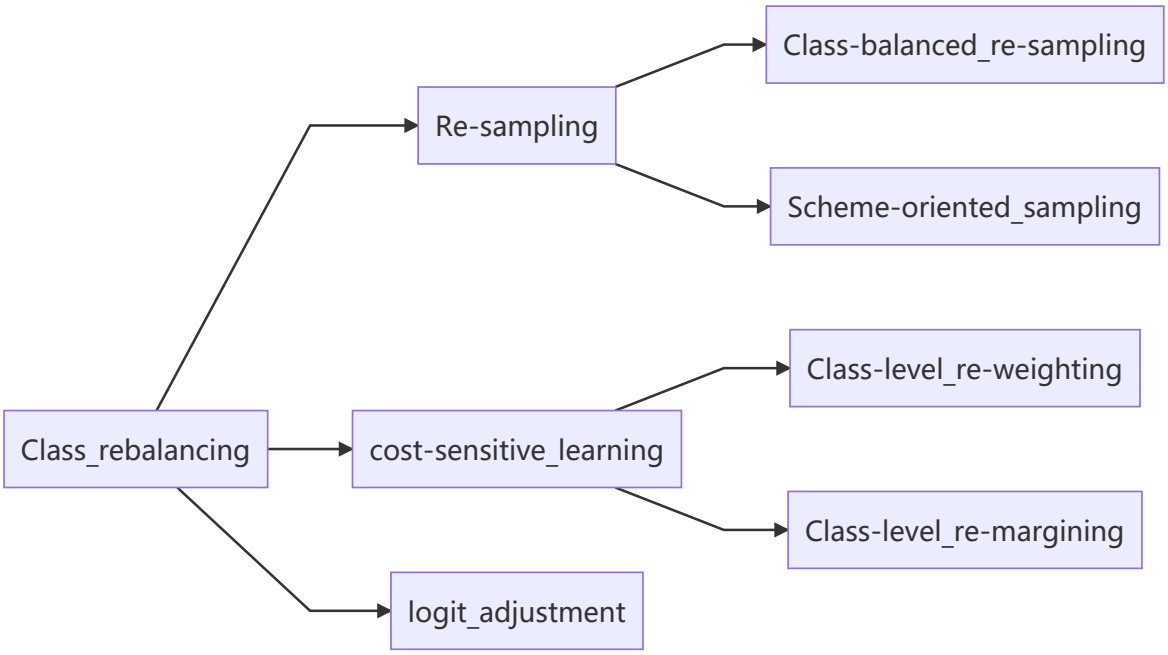
这篇文章总结的模型方法还是挺多的，分类也很细致，文章里提到的方法是在 mid-2021 前提出来的。在 Appendix 里面，我把论文里的分类用流程图展示出来了，每个方法有一些标签方便大家查阅。

我在 github 上也找到了长尾问题的 survey，里面有一直在更新新的长尾分布方法与代码，大家有兴趣也可以去看看。

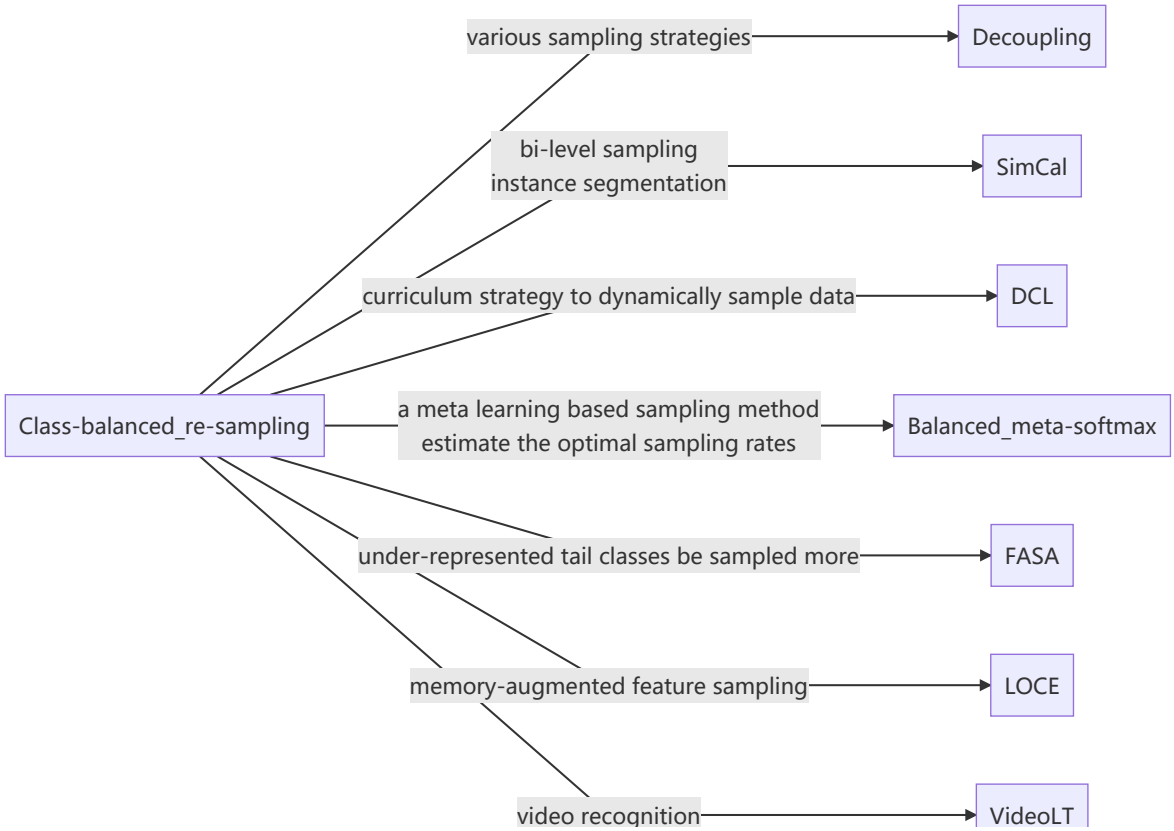
Appendix - CLASSIC METHODS

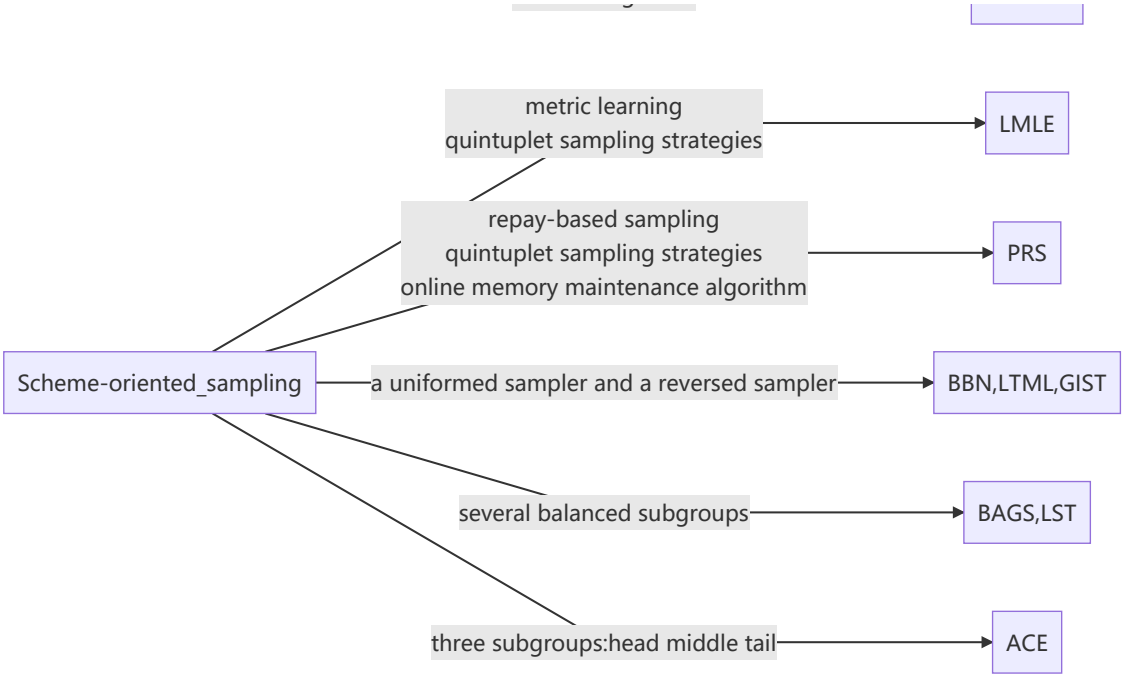


- **Class Re-balancing**

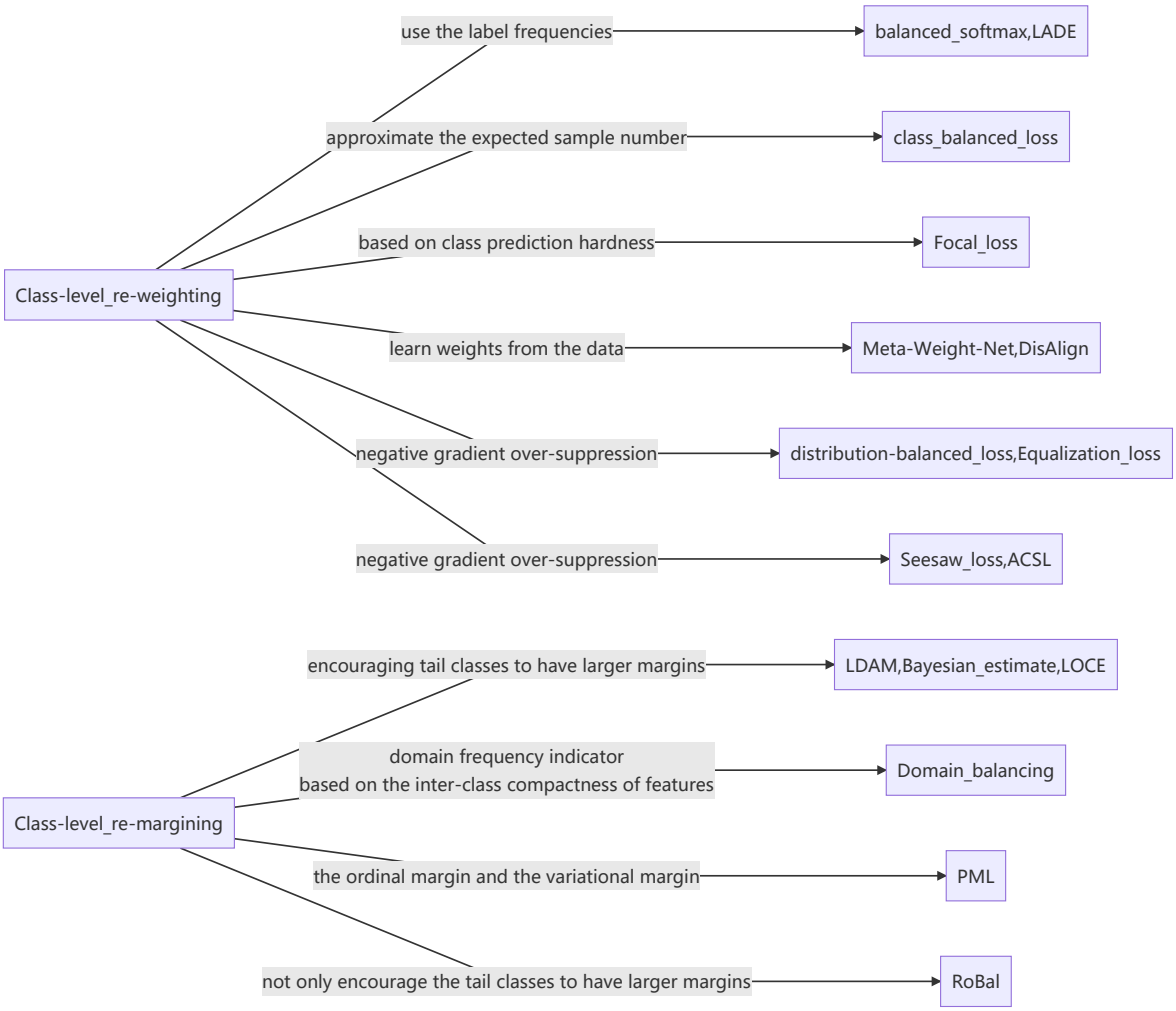

Class_rebalancing
Re-sampling
Class-balanced_re-sampling
Scheme-oriented_sampling
cost-sensitive_learning
Class-level_re-weighting
Class-level_re-margining
logit_adjustment

- **Re-sampling**

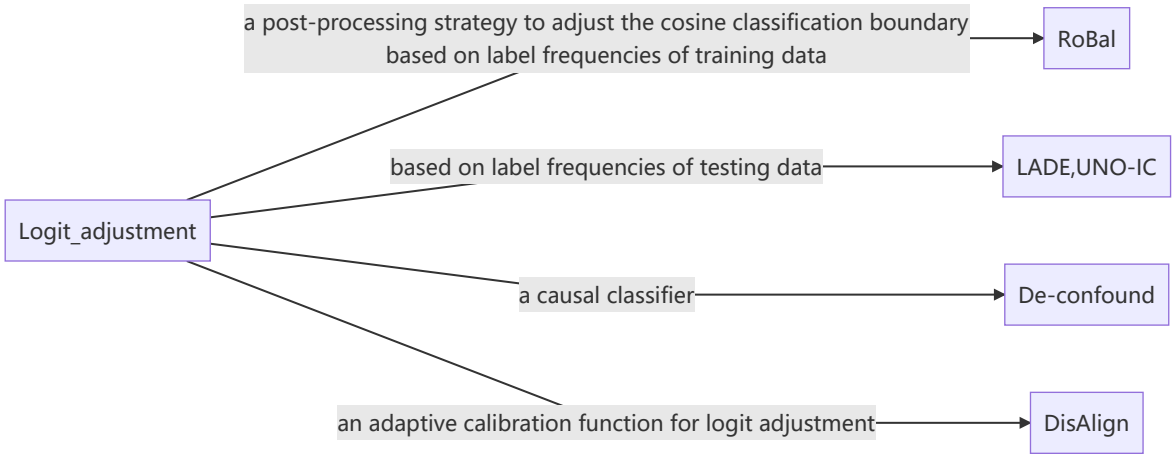

Class-balanced_re-sampling
various sampling strategies
Decoupling
bi-level sampling
instance segmentation
SimCal
curriculum strategy to dynamically sample data
DCL
a meta learning based sampling method
estimate the optimal sampling rates
Balanced_meta-softmax
under-represented tail classes be sampled more
FASA
memory-augmented feature sampling
LOCE
video recognition
VideoLT



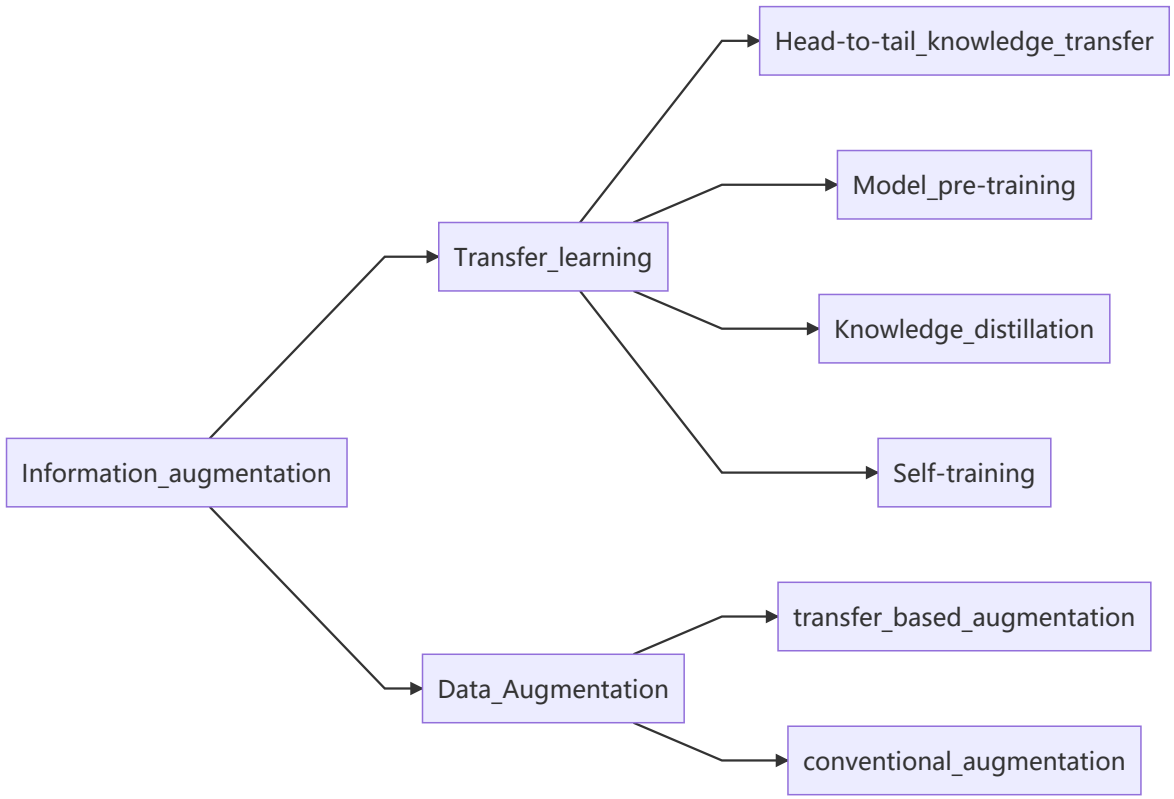
- **cost-sensitive learning**


Class-level_re-weighting
use the label frequencies
balanced_softmax,LADE
approximate the expected sample number
class_balanced_loss
based on class prediction hardness
Focal_loss
learn weights from the data
Meta-Weight-Net,DisAlign
negative gradient over-suppression
distribution-balanced_loss,Equalization_loss
negative gradient over-suppression
Seesaw_loss,ACSL
Class-level_re-margining
encouraging tail classes to have larger margins
LDAM,Bayesian_estimate,LOCE
domain frequency indicator
based on the inter-class compactness of features
Domain_balancing
the ordinal margin and the variational margin
PML
not only encourage the tail classes to have larger margins
RoBal

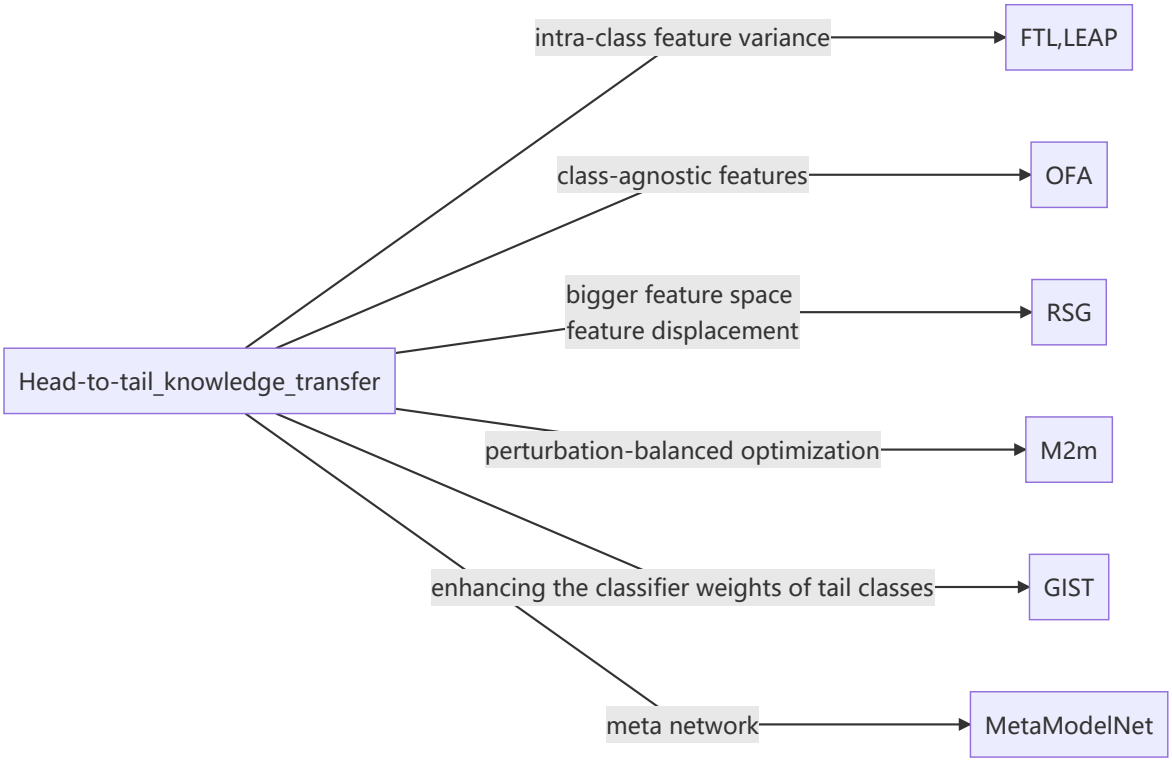
- **Logit adjustment**

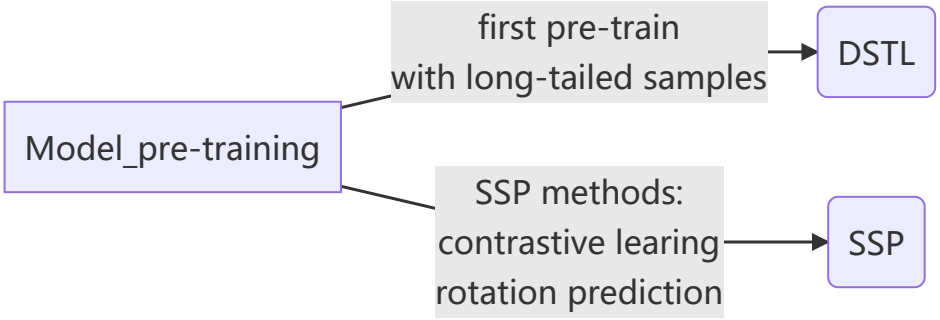


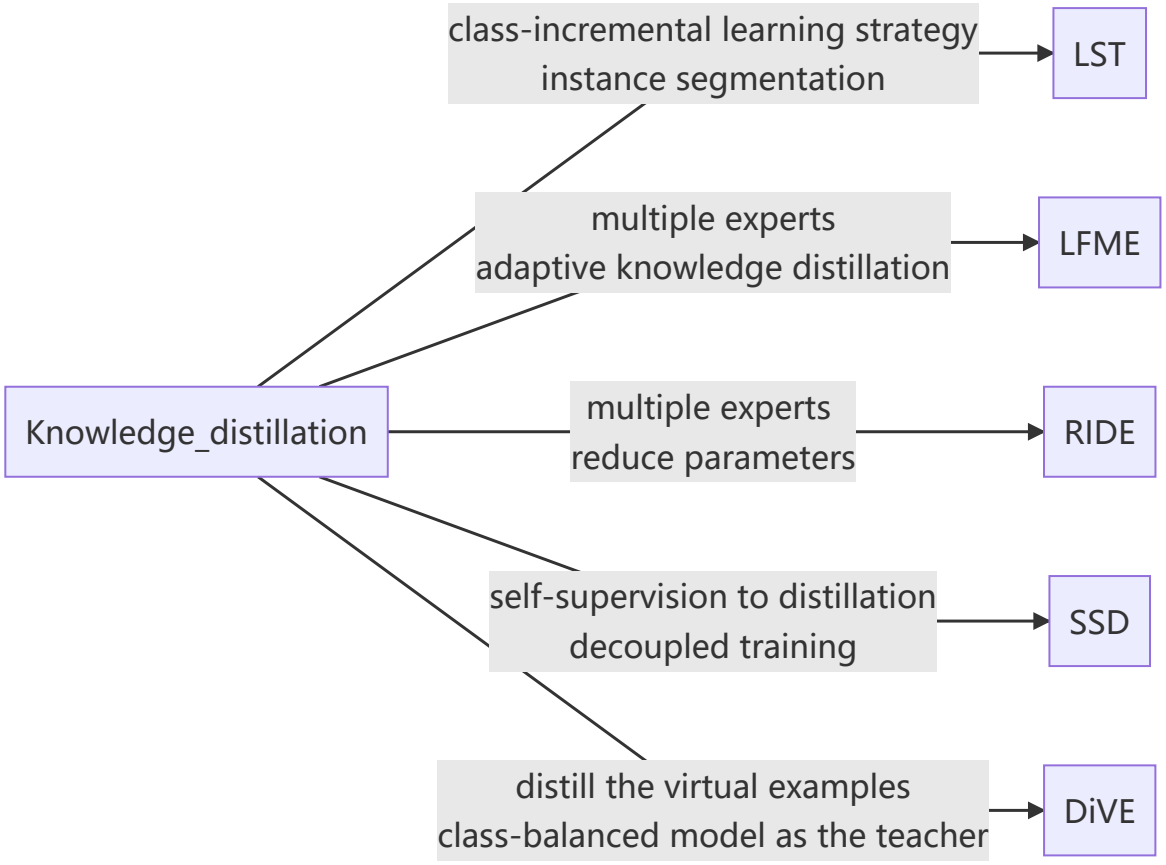
- **Information augmentation**

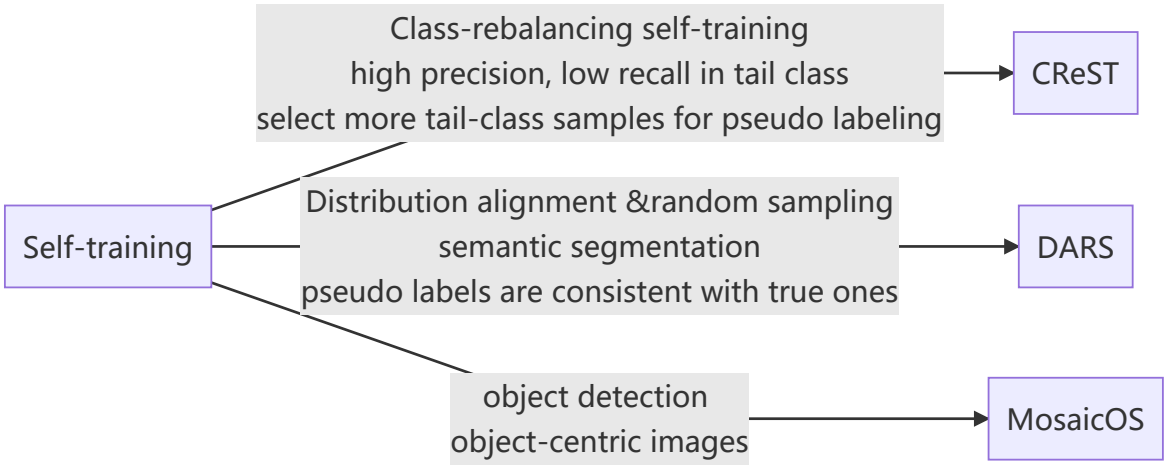


- **Transfer Learning**

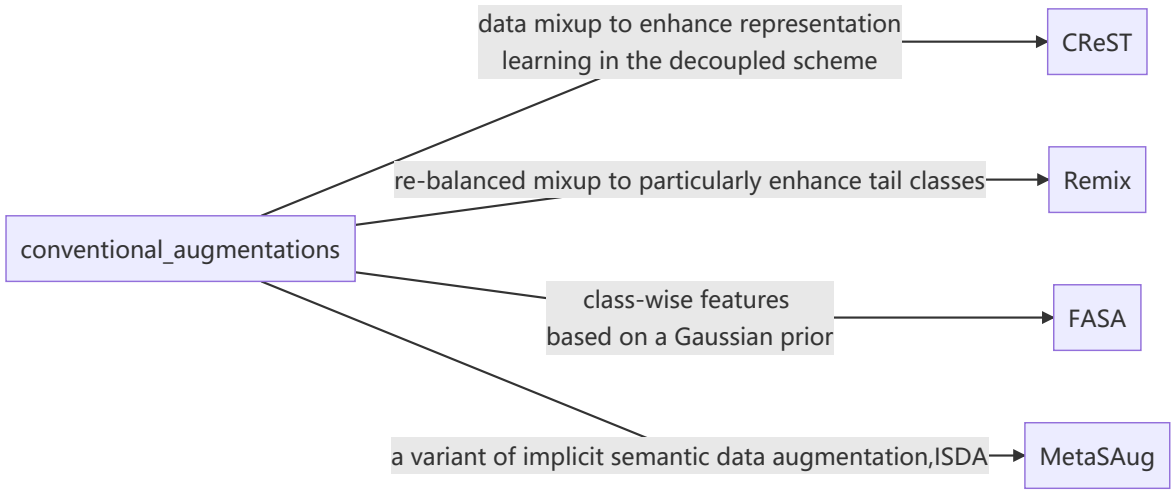




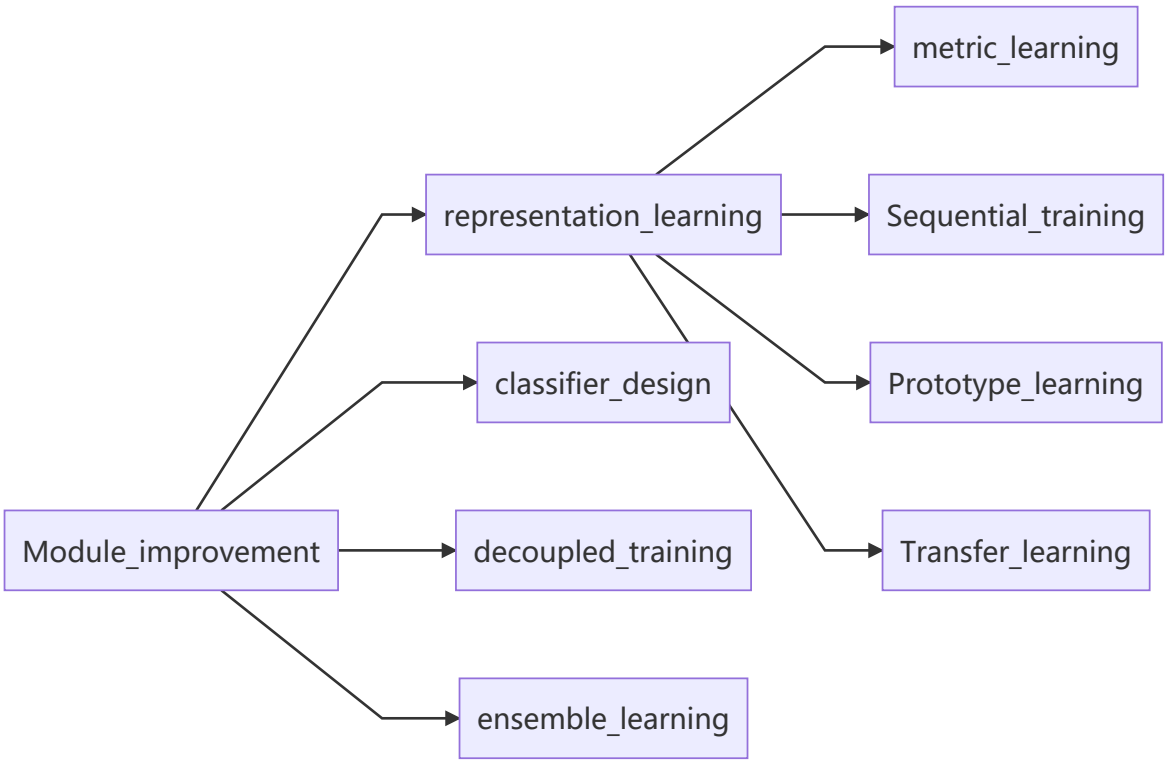




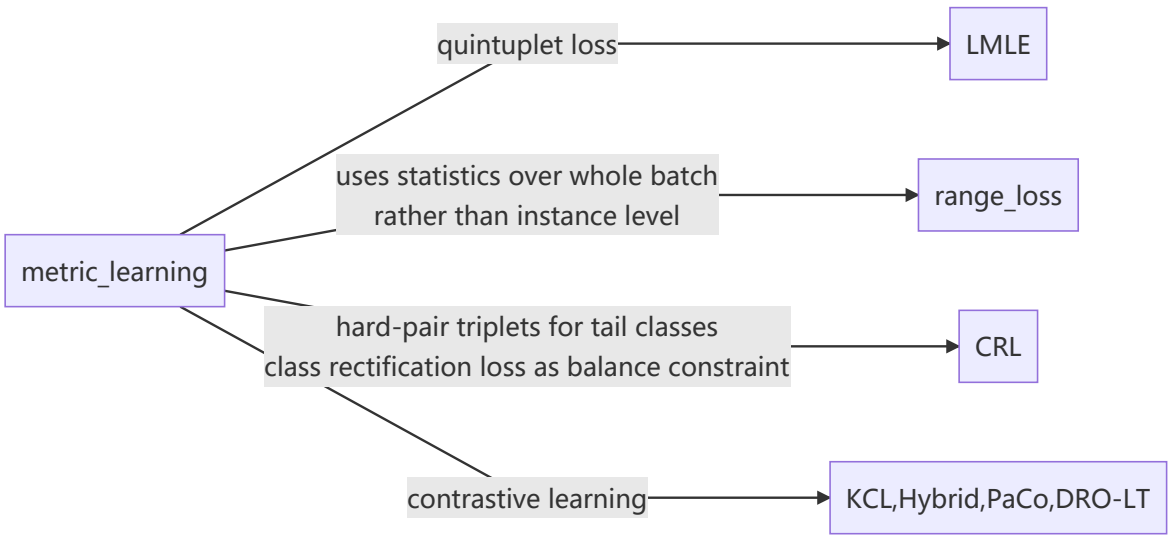
- **Data Augmentation**

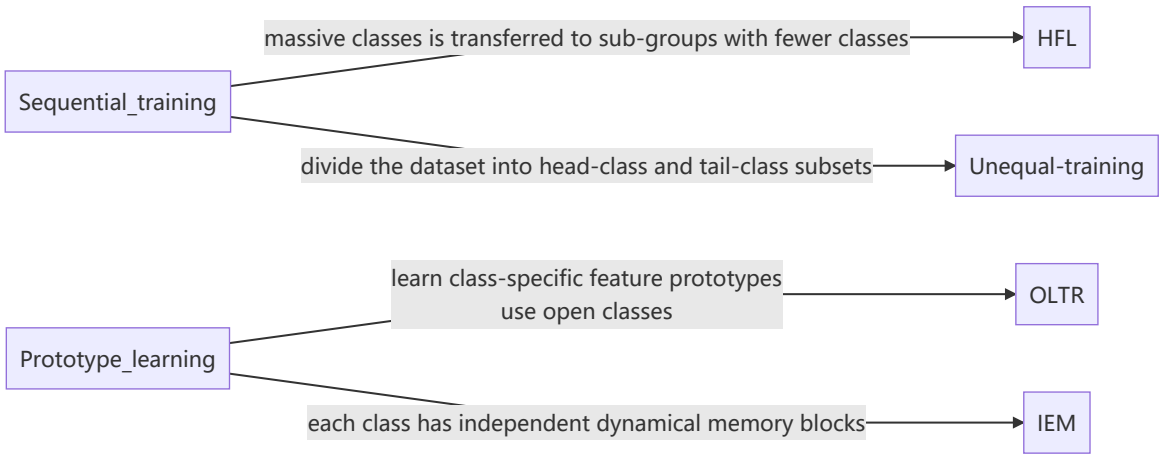


- **Module improvement**

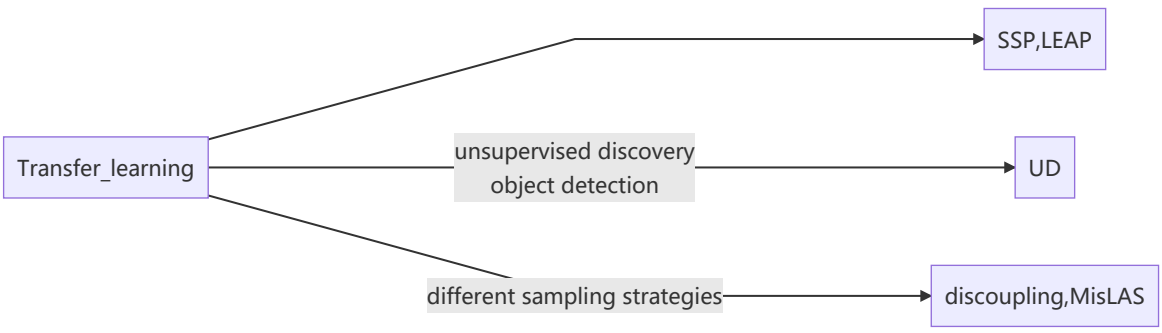


- **representation learning improves the feature extractor**

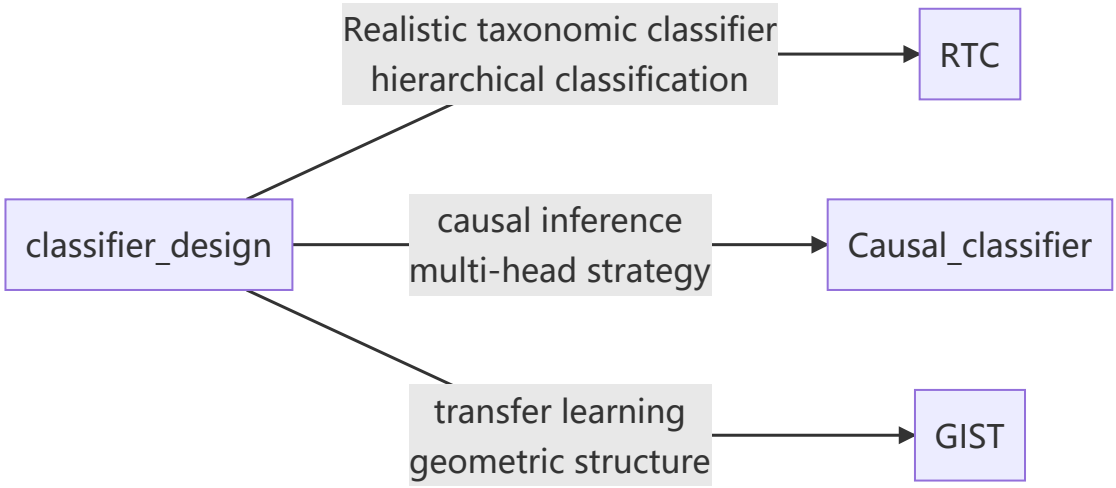

metric_learning
quintuplet loss
LMLE
uses statistics over whole batch
rather than instance level
range_loss
hard-pair triplets for tail classes
class rectification loss as balance constraint
CRL
contrastive learning
KCL,Hybrid,PaCo,DRO-LT


Sequential_training
massive classes is transferred to sub-groups with fewer classes
HFL
divide the dataset into head-class and tail-class subsets
Unequal-training

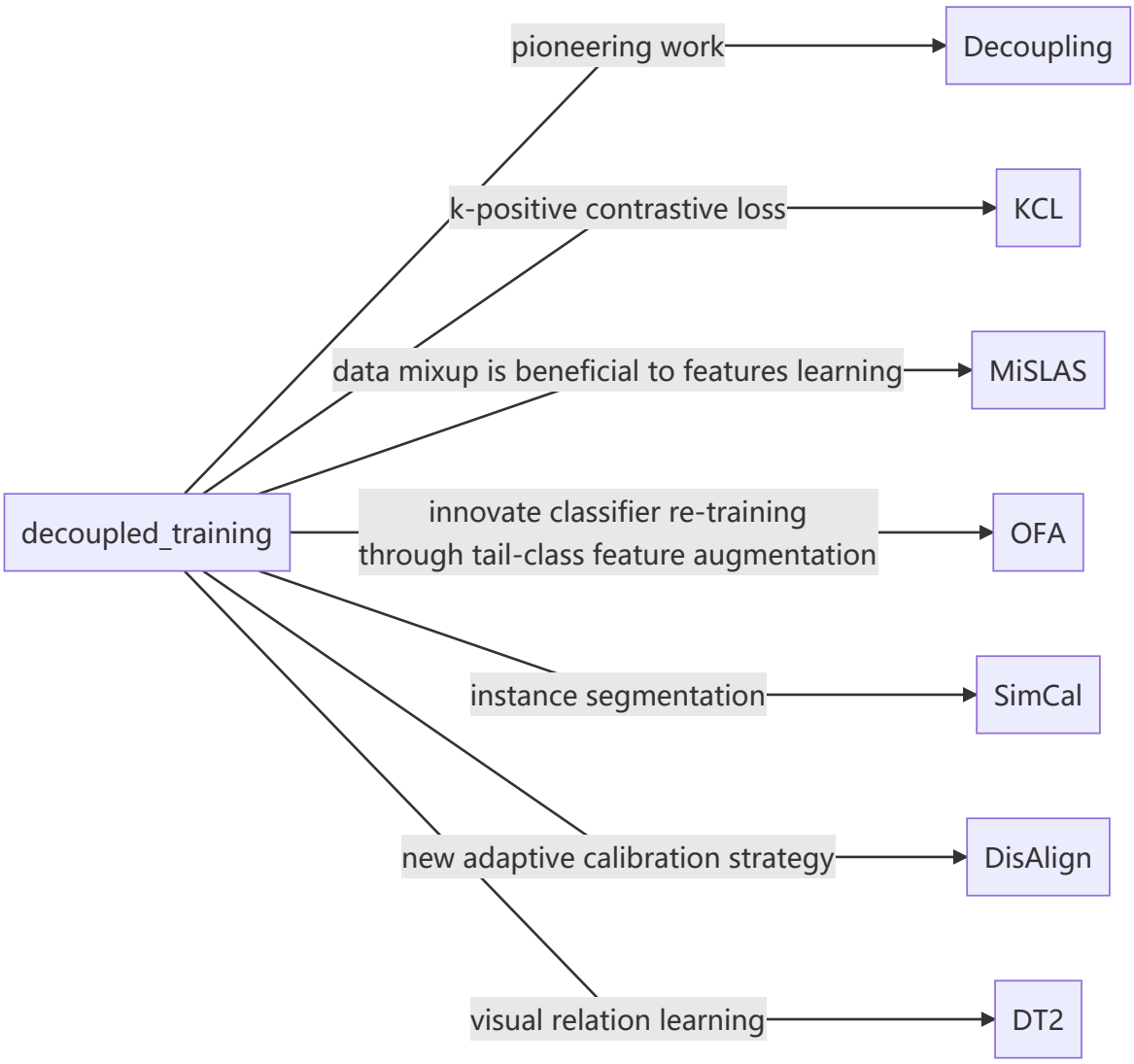
Prototype_learning
learn class-specific feature prototypes
use open classes
OLTR
each class has independent dynamical memory blocks
IEM


Transfer_learning
SSP,LEAP
unsupervised discovery
object detection
UD
different sampling strategies
discoupling,MisLAS

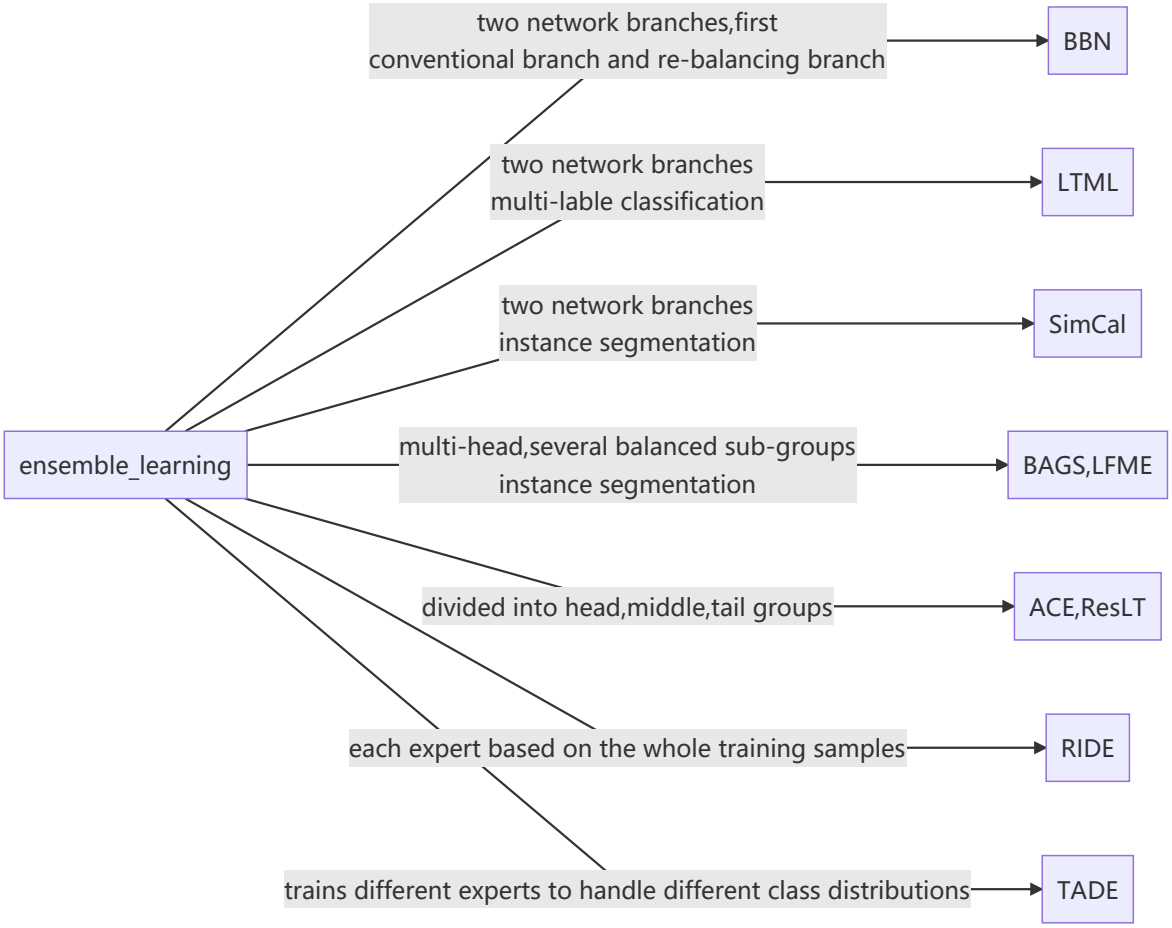
- **classifier design enhances the model classifier**



- **decoupled training boosts the learning of both the feature extractor and the classifier**



- **ensemble learning improves the whole architecture**


ensemble_learning
two network branches,first conventional branch and re-balancing branch
BBN
two network branches multi-lable classification
LTML
two network branches instance segmentation
SimCal
multi-head,several balanced sub-groups instance segmentation
BAGS,LFME
divided into head,middle,tail groups
ACE,ResLT
each expert based on the whole training samples
RIDE
trains different experts to handle different class distributions
TADE