点云语义分割领域自适应的多模态方法

12212635 韩骐骏　　12212553 侯雪冰　　12212961 欧阳莹轩

一. 引言

本项目的目标是应用迁移学习的方法帮助神经网络对点云进行语义分割。所以它涉及 3 个领域：多模态学习，点云语义分割，领域自适应。

很遗憾，我们最终没能做出一个成果，因此，本项目似乎被我们做成了像数学/统计/物理系开设的 seminar，同学们研读经典论文，却不产出成果。所以，这篇报告严格意义上说是一篇感悟，而非 tech report。但是说实话，我们并不觉得这个项目比其它项目带来的收获少，相反，通过对本项目相关知识的学习，我们发现自己以前对深度学习的掌握完全是皮毛，并且，通过查阅文献以及和学长交流，我们也逐渐了解了不少做深度学习研究的方法。下面，我们将展开说明自己在本项目中做了什么，学到了什么。

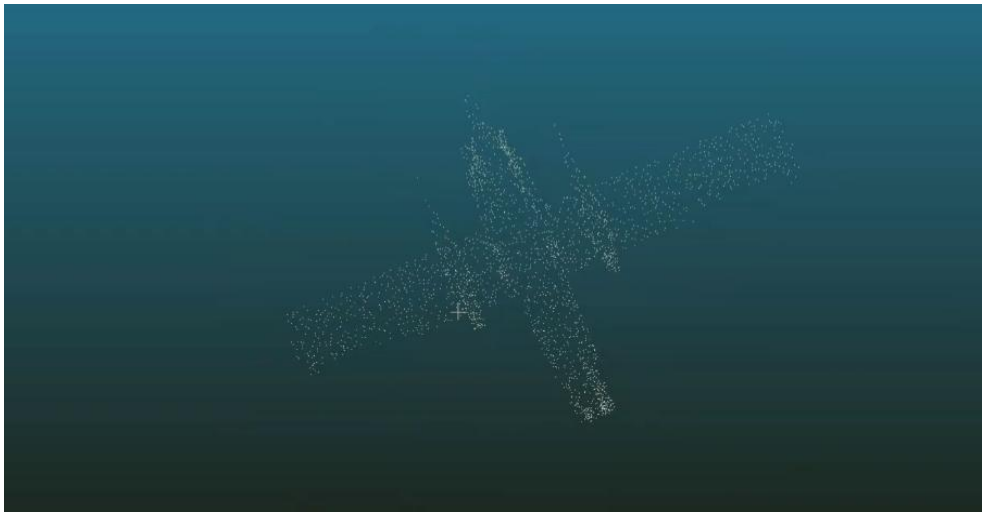
二. 项目基础

1. 点云处理

1.1 点云是什么?

点云是用来刻画三维世界的一种数据结构，或者说，就是某个坐标系下若干三维坐标组成的数据集合。点云数据具有无序性的特点，可分为有序点云和无序点云，它是一种无序、无结构的海量数据点的集合。点云数据具有高精度、高分辨率和高维度的几何信息，这是因为获取点云数据的测量设备一般有着极高的测量精度和较低噪声水平，密集分布的点可以详细描述数据背后的物理细节，除了空间的三维坐标外还可以包含颜色信息、强度信息等。

下图是一个示例



1.2 为什么要用点云?

对点云数据的使用需求基于计算机视觉相关的任务需求，比如对传感器采集的信息进行分类，分割等等。那为什么不用图像呢？因为实际物体是 3 维的，而图像始终是 2 维，因此 2D 图像一定无法呈现目标所有的信息，甚至，存在较大损失。而点云原本就是 3D 的，因此可以更好地描述空间中的物体。三维点云直接提供了三维空间的数据，而图像则需要通过透视几何来反推三维数据。

1.3 如何获取点云数据?

点云不能被普通的相机拍摄得到的，可以是通过三维成像传感器获得，比如双目相机、三维扫描仪、RGB-D 相机等。

点云可通过扫描的 RGB-D 图像，以及扫描相机的内在参数创建点云。相机校准可以获取相机内在参数如焦距、光学中心。根据所获数据计算真实世界的点 (x, y, z)。(u, v) 是图像坐标，(cx, cy) 是光学中心，(fx, fy) 是焦距，d 是深度值。

点云相比 RGB-D 图像有着更稀疏的结构。此外，获得点云的较好方法还包括 LiDAR 激光探测与测量，主要通过星载、机载和地面三种方式获取。

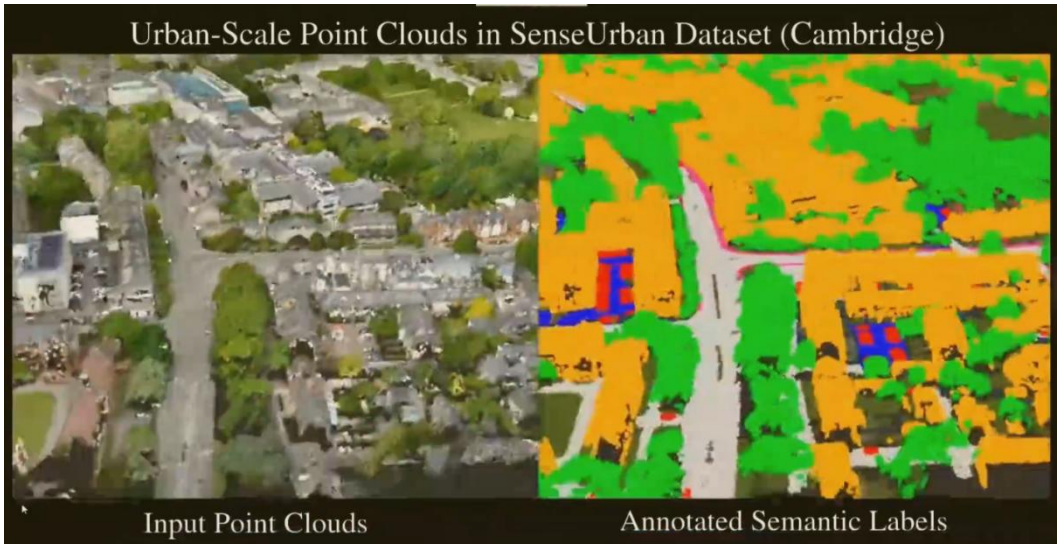
1.4 3D 和 2D 的转换

不过，有时也需要把 3D 的点云投影到一张 2D 图像上，也就是用一种映射方法把(x,y,z)变成另一个坐标系中的(x,y)，这是 3D 到 2D 的转换。注意，此时转换后的 2D 图像仍然保留了原始点云的全部信息，因为可以把它用相反的方式映射回去。

2. 语义分割

2.1 语义分割是干什么的?

语义分割从效果上就是：输入一幅点云图，然后网络输出的也是一幅点云图，不过每个点都有一个标签，指示其属于哪个类。一个直观的演示如下：

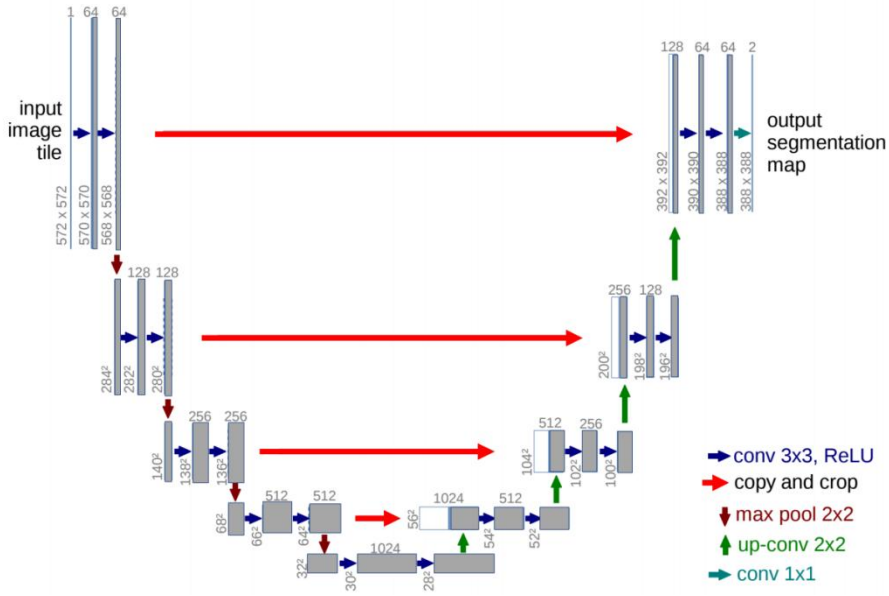


左侧是原始点云，右侧是分割后的点云。

2.2 典型的 2D 和 3D 语义分割 backbone

2.2.1 U-net

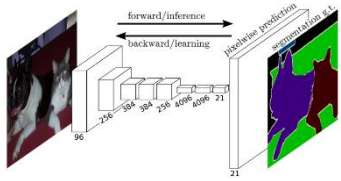
u-net 是典型的 encoder-decoder 模型，它的输入是一张 2D 图像，输出是一组通道数等于类数，分辨率等于输入图像的特征图。

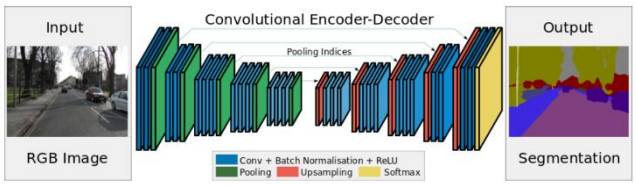


其工作原理为：

卷积层对输入图像进行连续的下采样，直至生成一个高度抽象的特征图，然后再把特征图不断进行上采样，并把 encoder 部分对应分辨率的特征图和 decoder 部分拼接，直至分辨率和原始图像相同。最后一步通过 1　1 卷积把每个通道最佳的类提取出来，变成一个单通道的分割结果。

事实上，很多语义分割的网络都是 encoder-decoder 架构，先提取特征，再进行上采样，下图分别为 FCN 和 segnet。





可见，这几个分割网络的 encoder 部分大都采用与 VGG 类似的结构，decoder 则有所不同，FCN 和 unet 使用了反卷积层，segnet 则采用了反池化。并且，unet 采用了 copy and crop 技术使得 encoder 中丢失的信息在解码器中能得以应用（应该是残差的思想）。

2.2.2 稀疏卷积(SparseConvNet)[4]

2.2.2.1 为什么使用稀疏卷积

稀疏卷积常用于 3D 项目中。3D 点云数据的稀疏性导致其无法使用标准的卷积操作，稀疏的 3D 点云数据通常包含很多零值，而在标准的卷积操作中，卷积核在滑动时会覆盖输入数据的所有位置，零值也会被考虑进去，导致有效信息丢失，计算效率低下。故需要通过建立哈希表，保存特定位置的计算结果，实现稀疏卷积。

2.2.2.2 稀疏卷积的基本概念和实现

稀疏卷积类似稀疏矩阵操作，当矩阵中只有少数非零元素，即过于稀疏时，使用专门的数据结构存储这些非零元素及其位置，以减小内存需求并加速运算。

在稀疏卷积网络中， 输入张量通常是一个多维数组，其中包含了输入数据的特征表示。对于图像数据而言，输入张量通常是一个四维数组，其维度依次表示批次大小（batch size）、通道数（channels）、高度（height）、宽度（width）。包含空间和时间维度（长、宽、高、时间等）以及一个特征空间维度（是描述数据特征的空间的维度数）。

对输入数据进行稀疏格式的转换，记录非零值的坐标及非零值具体值，卷积核也同样表示为稀疏格式，只包含非零值及其位置。卷积操作时，将稀疏卷积核中的非零值与输入张量的非零值位置对齐并进行点积。只有在输入张量和卷积核的非零值位置重叠时，才进行计算。将所有计算结果进行合并，生成输出张量。输出张量同样以稀疏格式存储，仅记录非零值及其位置。

3. **半监督学习 (Semi-supervised learning)**

3.1 Pseudo-Label 模型(2013)

其核心思想包括两个步骤：1、运用训练出的模型对无标签数据进行预测，以概率最高的类别作为无标签数据的伪标签。2、运用熵正则化，将无监督数据转为目标函数的正则项。将拥有伪标签的无标签数据视为有标签的数据，然后用交叉熵来评估误差大小。

模型整体的目标函数：

$$L=\frac{1}{n}\sum_{m=1}^{n}\sum_{i=1}^{C}L(y_i^m,f_i^m)+\alpha(t)\frac{1}{n'}\sum_{m=1}^{n'}\sum_{i=1}^{C}L(y_i'^m,f_i'^m)$$

其中红框外的部分是标签数据集的交叉熵损失，n表示有标签样本的数量，C是类别的数量，y_i^m表示真实标签，f_i^m表示类别i的预测概率。当模型预测和真实标签偏离时，交叉熵损失会增大，在参数的更新调整中减少预测误差。

红框内的部分是无标签数据集的熵正则化：n'是无标签样本的数量，y_i^m是伪标签，$f_i'^m$是无标签样本m的类别i的预测概率。$\alpha(t)$是一个随时间（迭代次数）增加的权重因子。

$$\alpha(t)=\begin{cases}0 & t<T_1\\ \frac{t-T_1}{T_2-T_1}\alpha_f & T_1\le t<T_2\\ \alpha_f & T_2\le t\end{cases}$$

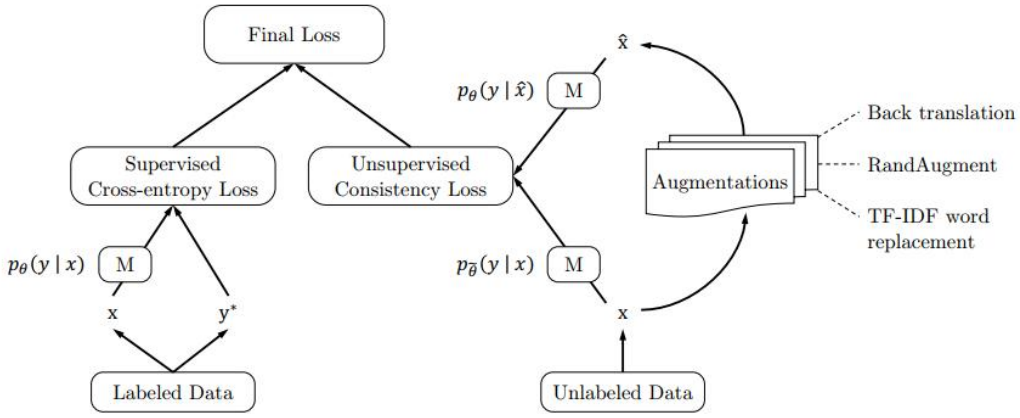
红框内的这部分通过将高置信度的预测当作真实标签来帮助模型从无标签数据中学习，把高置信度的伪标签样本纳入损失函数中，计算它们的交叉熵损失。这一部分损失称为熵正则化项利用无标签数据上的高置信度预测提取额外的训练信号，即用无标签数据增加训练数据量，提高模型在未见过的测试数据上的泛化能力。

3.2 UDA (Unsupervised Data Augmentation for Consistency Training2019)

UDA 算法通过结合有标签和无标签数据进行训练实现半监督学习，提高模型性能。核心思想是利用数据增强技术来生成无标签数据的不同版本，并通过一致性损失来引导模型在增强后数据上进行的一致性预测。

3.2.1 有监督学习：

输入 是有标签数据，模型预测$p_\theta(y|x)$是模型对输入数据的预测输出。根据预测值和真实标签进行交叉熵损失的计算，用于指导有标签数据的训练。



3.2.2 无监督学习：

输入 是无标签数据，通过数据增强在保证语义不变的前提下使有标签数据的表现形式得到丰富。图中列出了常用的数据增强技术：回译 Back translation（将文本翻译再翻回），RandAugment（图像增强技术如旋转、缩放、颜色调整等）和 TF-IDF 词替换。增强后预测$p_\theta(y|\hat{A})$是模型对增强数据的预测输出。一致性训练（Consistency Training）和一致性损失（Unsupervised Consistency Loss）是利用增强后的无标签数据，通过计算增强前后模型输出差异来训练。采用 KL 散度或者均方误差（MSE）来预测分布之间的差异，这部分损失引导模型再增强前后保持一致的预测，充分利用无标签数据的潜在信息。

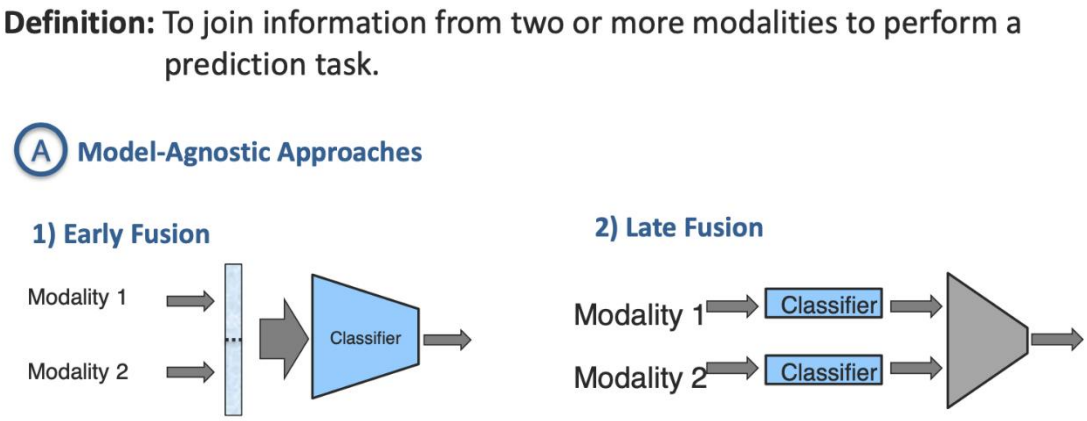
4. 多模态学习 (MultiModal Machine Learning (MMML))

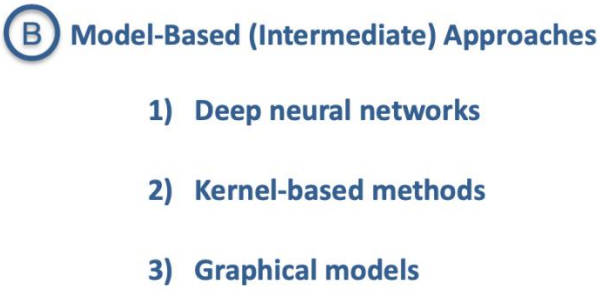
4.1 定义

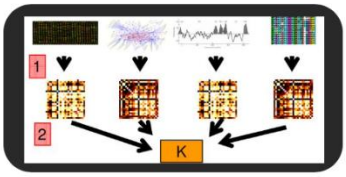
模态（modal）是表达或感知事物的方式，每一种信息的来源或者形式，都可以称为一种模态，在两种不同情况下采集到的数据集亦可认为是两种模态。

4.2 特征融合 (Fusion)

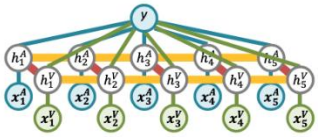
结合来自多个模态的信息来执行预测。与模型无关的方法有早期融合、晚期融合、混合融合。基于模型的方法有深度神经网络、多核学习、图模型。







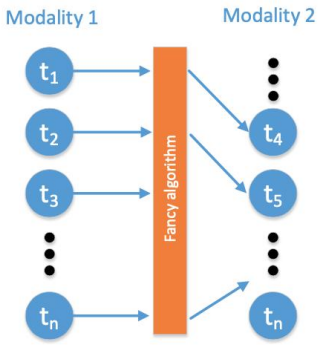
Multiple kernel learning

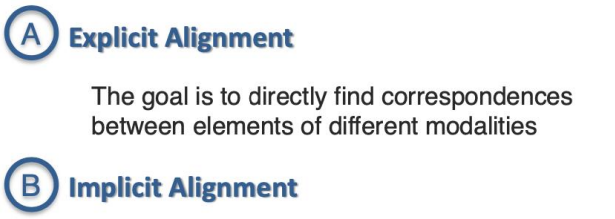

Multi-View Hidden CRF

4.3 **特征对齐**（Alignment）

从多种不同模态中识别子元素之间的直接关系。**显式对齐**是通过相似性度量（欧氏距离、余弦相似度等），将不同模态的子组件在统一的表示空间中进行匹配。**无监督对齐**在训练过程中不依赖预先标注的对齐结果，模型必须同时学习相似度度量和对齐方式，根据两个模态的数据输入，通过数据内在结构和分布进行自我监督，学习如何对其不同模态的子元素。**有监督对齐**利用带有明确对齐关系标注的训练数据进行学习，实现相似度度量和对齐方式，比无监督对齐更加准确。特征对齐的作用在于其对匹配性的操作可以更有效地融合多模态信息，让模型更好地理解和运用信息，提高准确性。

Definition: Identify the direct relations between (sub)elements from two or more different modalities.

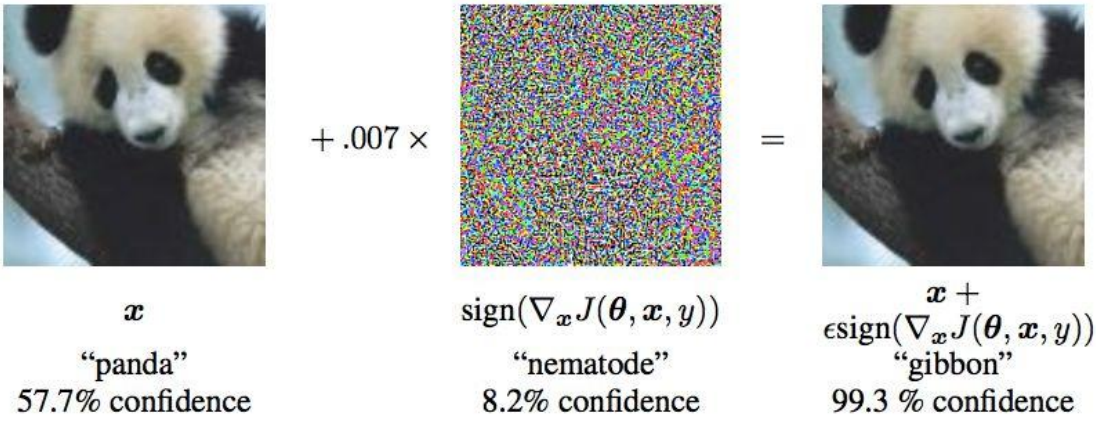




1. **对抗式训练**

深度学习中的对抗有多重含义，一者是 GAN 这一生成模型，另一者“对抗训练”主要关心模型在小扰动下的稳健性。对模型进行对抗攻击，可以使模型增强防御，提升

稳健性能，更不容易出错。对抗训练需要“对抗样本”



上图最左侧生成的图像就是对抗样本。最左侧的图像预测结果是模型以 57.7%的置信度将其分类为熊猫，但是经过添加中间的扰动（通过计算损失函数相对于输入图像的梯度，然后取其符号（sign）得到的。J(θ ,x,y) 是损失函数， θ 是模型参数，x 是输入图像，y 是正确的标签),最左侧的图像对人类来说没有看起来的变化，可是模型以 99.3%的置信度将其错误分类为长臂猿。对抗样本，即在原始输入的基础上添加了设计的微小扰动，大幅改变模型输出结果。对抗样本暴露了模型的脆弱性和特定情况下的可靠问题。“对抗攻击”即制造更多对抗样本，“对抗防御”即通过优化使得模型能正确识别更多的对抗样本。

FGM 对抗训练方法（Fast Gradient Method）：对抗训练可以表示为一个 Min-Max 优化问题，如公式：$\min_{\theta} E_{(x,y)\sim D}\left[\max_{\Delta x \in \Omega} L(x+\Delta x, y; \theta)\right]$ 公式表示在最坏情况（Δx扰动最大）下，最小化期望损失，其中θ是模型参数，$L(x+\Delta x, y; \theta)$是模型损失函数，Ω是扰动的限制范围。为解决此问题，分内外两个循环进行优化。内循环：对每个样本（x，y），寻找最大化损失$L(x+\Delta x, y; \theta)$的扰动Δx。外循环：最小化最大扰动下的损失。而 FGM 方法可以快速生成对抗样本，通过计算损失函数的梯度来确定扰动方向，公式：$\Delta x = \epsilon \cdot \text{sign}\left(\nabla_x L(x, y; \theta)\right)$，公式中$\epsilon$是一个微小常数，控制扰动幅度。sign 表示取梯度的符号函数。根据公式，可以从输入样本 x 计算出损失函数相对于 x 的梯度，再又梯度生成扰动Δx，将其添加到原始输入 x 上得到对抗样本$x + \Delta x$，用其进行训练，最小化损失。

2. **KL 散度与交叉熵**

对于一个随机变量 X，已知它的概率分布为 p(x)，那么可以用信息熵来衡量 X 在整个概率空间信息量的期望，即

$$H(x) = -\sum_{x} p(x) \log\left(p(x)\right)$$

其实，1/log(p(x))这个因子正是信息量，概率越小信息量越大。而这里我们直接把信息量对 X 求期望，得到的其实是“客观上”的期望。那么，可以不可以求一个“主观”的期望呢？其实是可以的，在机器学习中，至少在预测任务中，我们的目标实际上就是让“主客观”尽量一致。

比如 X 在我的主观经验中有这样一个分布 q(x)，而 ground truth 是 p(x)，那么我们可以用下式衡量这种主客观的差异：

$$H\left(p(x), q(x)\right) = -\sum_{x} p(x) \log\left(q(x)\right)$$

这就是交叉熵。可以证明，交叉熵的下界是信息熵。
主客观差异越大，交叉熵越大，反之越小。当 p(x)=q(x)时，交叉熵退化为信息熵。
不过，可以发现，即使 p(x)=q(x)，交叉熵也不为 0，那直接把交叉熵减掉一个信息熵，就可以实现这个效果，这就是 KL 散度（Kullback—Leibler divergence）。

$$D_{KL}(p||q) = H(p,q) - H(p) = \sum_x p(x) log \frac{p(x)}{q(x)}$$

KL 散度也被称为相对熵，通过前面的讨论，可知 KL 散度一定是非负的。
在信息论中，D（p||q）是被这样理解的：本来你应该以概率分布 p 为基准对数据进行编码的，结果你误以 q 为基准编码了，那么这样做你就需要额外的信息来进行编码，这个“额外的信息”就是 D(p||q)。在机器学习语境中，可以把 p 看作外界的视角，而把 q 当作主观视角，如果这两种视角完全一致，那么 D(p||q)=0。在实际的优化过程中，由于信息熵那一项和自变量无关，所以把交叉熵和 KL 散度作为目标函数进行优化的结果是相同的。

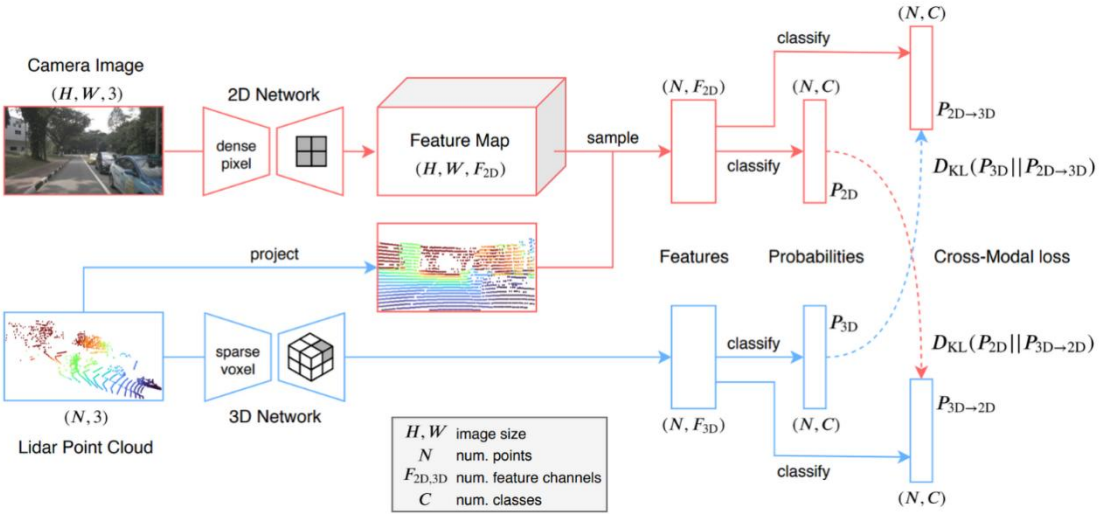
三. 论文阅读

1. xMUDA[5]

1.1 目的

现在已经有 3D 网络能直接对点云进行分割了，但是 3D 网络对一些领域的变化比较敏感（如白天到黑夜），而 2D 则更鲁棒。因此可以利用两种不同模态之间的互补信息来增强模型的鲁棒性。

1.2 总体架构



首先，数据集分为两部分，带标签的源域和不带标签的目标域。其中，源域有 2D 图像（H,W,3）和与之匹配的 3D 点云（N，3），3D 点云带有标签，而目标域只含有 2D 图像和对应的 3D 点云，却不带标签。
这个网络有两条数据流，一条是 2D 网络特征提取，另一条是 3D 网络特征提取。

2D 图像先经过 2D 分割网络得到一组(H,W,F_{2D})的特征图，同时，把 3D 点云图投影到一张图像上，对比投影的图和特征图，找到投影图上对应的特征位置，这一步就是把 3D 特征与 2D 特征对齐。然后再对特征图进行采样，也就是把所有 3D 点在 2D 特征图中提取出的特征向量组合起来，形成一个新的特征矩阵，这个矩阵的每一行代表了一个点

的特征向量。最后，该特征矩阵被分类器处理，输出尺寸为(N, C)的预测值P_{2D}，其中 C 代表类别数，这也就是每个点所属类的概率分布，这里使用的损失函数是预测值与真实值之间的交叉熵。

$$\mathcal{L}_{seg}^{2D}(x_s, y_s) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} y_s^{(n,c)} \log P_{2D}^{(n,c)}$$

另一边，3D 网络直接对输入的点云图进行处理，得到对应的 3D 特征矩阵，然后再通过另一个分类器得到P_{3D}。直到此处，我们使用的都完全是监督学习。

$$\mathcal{L}_{seg}^{3D}(x_s, y_s) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} y_s^{(n,c)} \log P_{3D}^{(n,c)}$$

现在，关键的地方来了，P_{2D}和P_{3D}的维数是一样的，那么它们每个点的概率所属类的概率分布也一样吗？注意到，前者是由 2D 图像作为输入得到，而后者则直接从原始点云提取特征得到，由于 2D 图像的信息和 3D 点云是不同的，所以得到的P_{2D}和P_{3D}也不同。那么，我们必须设计一种方法，使得P_{2D}和P_{3D}的差异尽量小，这样，即使是以 2D 图像作为输入，得到的分割结果和从 3D 网络生成的区别不大，反过来，当输入为 3D 点云（如黑夜中的场景）时，你甚至不需要标签，只要有对应的 2D 图像，也可以得到比较好的分割效果。

那么，怎么实现这一点呢？先说 2D 网络，我们保持分类器不变，但是新增一个分割头，它的损失函数不再仅仅是预测与标签的交叉熵，而是预测与 3D 网络预测结果的 KL 散度。在训练时，我们把总的损失函数定义为两者的线性组合，这样就能使得 2D 网络不仅能预测 2D 图像的分割结果，还能预测 3D 网络的分割结果!

$$\mathcal{L}_{xM}^{2D}(x) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} P_{2D}^{(n,c)} log \frac{P_{2D}^{(n,c)}}{P_{3D}^{(n,c)}}$$

同样地，再在 3D 网络的分类器中增加一个分割头，让它也能预测 2D 网络的分割结果，这样就能充分利用模态间的互补信息。

$$\mathcal{L}_{xM}^{3D}(x) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} P_{3D}^{(n,c)} log \frac{P_{3D}^{(n,c)}}{P_{2D}^{(n,c)}}$$

那么问题来了，为什么不直接把新增的分割头放在整个预测结果的后面，而是放在输出的倒数第二层呢？这是因为，如果直接放在最后，那么得到的将是分类器取最大值后的结果，这样的信息显然不如在每个通道上更丰富。

最终，2D 网络的损失函数为

$$\frac{1}{N_s}\sum_{x_s \in S}(\mathcal{L}_{seg}^{2D}(x_s, y_s) + \lambda_{s\mathcal{L}_{xM}^{2D}}(x_s)) + \frac{1}{N_t}\sum_{x_t \in T}\lambda_{t\mathcal{L}_{xM}^{2D}}(x_t)$$

3D 网络的损失函数为

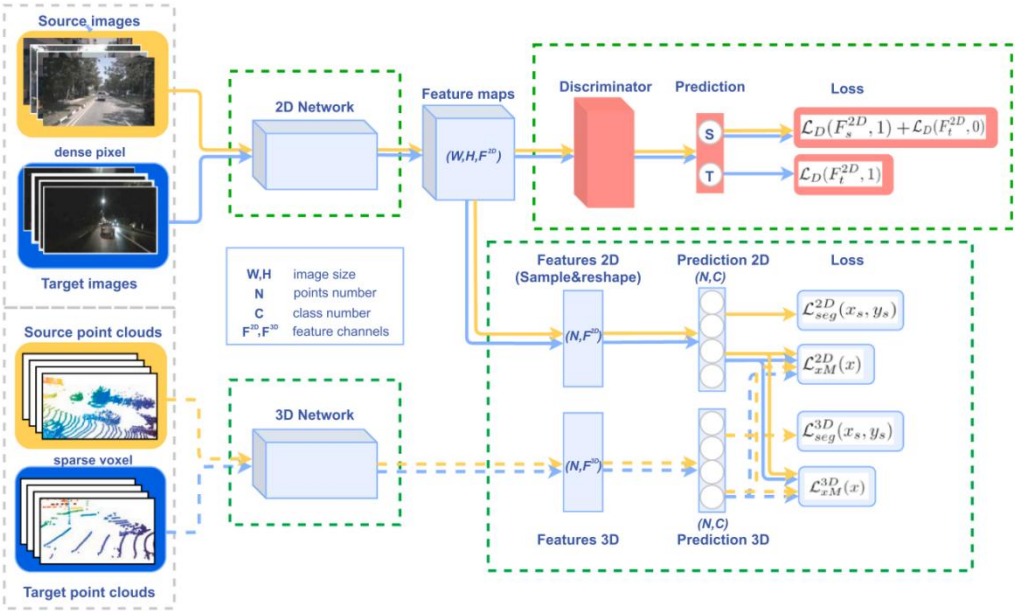
$$\frac{1}{N_s}\sum_{x_s\in S}(\mathcal{L}_{seg}^{3D}(x_s,y_s)+\lambda_{s}\mathcal{L}_{xM}^{3D}(x_s))+\frac{1}{N_t}\sum_{x_t\in T}\lambda_{t}\mathcal{L}_{xM}^{3D}(x_t)$$

1.3 实现细节

2D 部分的 backbone 是一个 unet 类型的网络，不过作者做了一些修改，在 encoder 部分他改为了 resnet34，这个做法来源于[6]。3D 部分则是一个 sparseconvnet。在训练时，2D 和 3D 网络是一同训练的。

2. AUDA[7]

2.1 总体架构



可以看到，这篇文章提出的模型架构和 xMUDA 其实非常相似，其实本文作者也是以 xMUDA 作为 baseline 改进的。其最大的不同在于两点：第一，提取 2D 特征的方法；第二，损失函数的类敏感化。

先说第一点。在 xMUDA 中，2D 特征是通过把点云投影与 2D 特征图比较后采样得出的，本篇作者认为，这样的做法会产生如下问题：由于 2D 特征在投影图上的像素非常稀疏（只有一小部分点能和 2D 图像的特征相匹配），所以其实这种采样只利用了很少的点，而没有利用整张图片的结构信息，导致 2D 分割网络的性能不够好。那怎么更好地利用 2D 图像的全部信息，从而获得更好的性能呢？这里作者使用了对抗式训练的方法，即在 2D 分割网络的后面加一个判别器，判别器接收 2D 网络的特征图并判断它是来自源域还是目标域。具体的做法是，把来自源域的图像标记为 1，把来自目标域的图像标记为 0，然后用一个交叉熵来衡量判别器对域进行分类时产生的误差大小。那么判别器的目标函数为：

$$\min_{\theta_{Dis}}\frac{1}{N_t}\sum_{x_t^{2D}}cL_D(F_t^{2D},0)+\frac{1}{N_s}\sum_{x_s^{2D}}cL_D(F_s^{2D},1)$$

2D 网络则同时在源域和目标域上训练，并尽量迷惑判别器，避免其识别出特征图来自哪个域，这样，就实现了更好的 UDA 性能。所以，在训练 2D 分割网络时，还要在目标函数中加上一个对抗项$\min_{\theta_{2D}} \frac{1}{N_t} \sum_{x_t^{2D}} cL_D(F_t^{2D}, 1)$。

此外，点云数据有一个特点，那就是不同类别占有的点的比例是不同的，有时差别非常大。所以针对一些比较稀少的类，有必要做特殊的处理。也就是，属于某一类的点越稀少，训练时应该给这样的样本赋予更大的权重，否则在数量劣势下这样的点就被埋没了。具体实现如下：

$$w_c = \frac{\log\left(\frac{\alpha \sum_{k=1}^{C} n_k}{n_c}\right)}{\min_{j \in \{C\}} \log\left(\frac{\alpha \sum_{k=1}^{C} n_k}{n_j}\right)}$$

其中　表示第 i 个类的点的个数，C 为类的总数，α 为一个超参，控制这种操作的影响程度，　　表示第 c 个类的样本的权值。这个思路来源于[8]。

所以，我们把这个权值放在损失函数中，就能更好地处理类之间的不平衡问题。

$$\mathcal{L}_{seg}^{2D}(x_s, y_s) = -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} w_c \, y_s^{(n,c)} \log P_{2D}^{(n,c)}$$

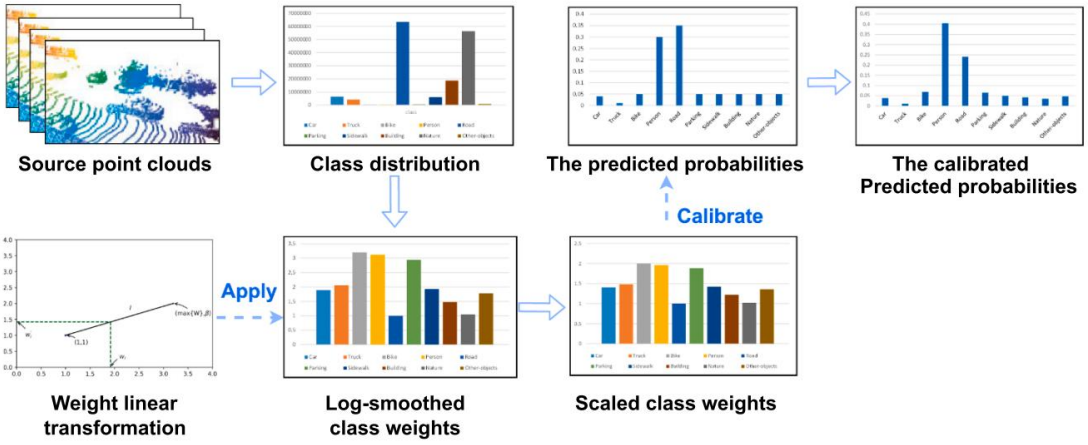
$$\mathcal{L}_{seg}^{3D}(x_s, y_s) = -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} w_c \, y_s^{(n,c)} \log P_{3D}^{(n,c)}$$

而在测试集上，也使用了类似的映射操作使得类的分布更平衡。

2.2 实现细节

2D 和 3D 分割网络和 xMUDA 用到的完全一致，而对抗网络则使用了 4 层全卷积网络。目标函数使用 Adam 优化器进行优化。

2.3 实验

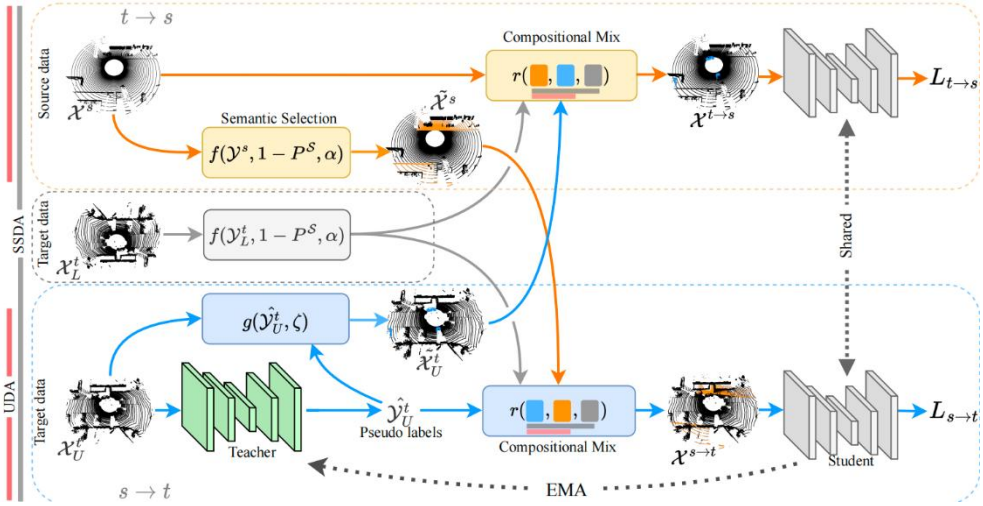


经过这两点改进，AUDA 在同一个 benchmark 上超过了 xMUDA。

Table 2
Comparisons of different models in three different UDA scenarios (%) in terms of mIoU. For brevity, we use SG for Singapore and SK for SemanticKITTI. *Avg* takes the mean of the predicted 2D and 3D probabilities after softmax.

Method	USA → SG			Day → Night			A2D2 → SK		
	2D	3D	Avg	2D	3D	Avg	2D	3D	Avg
xMUDA (Jaritz et al., 2020)	59.3	52.0	62.7	46.2	44.2	50.0	38.3	46.0	44.0
$xMUDA_{PL}$ (Jaritz et al., 2020)	61.1	54.1	63.2	47.1	46.7	50.8	41.2	**49.8**	47.5
AUDA (ours)	59.8	52.0	63.1	49.0	47.6	**54.2**	43.0	43.6	46.8
$AUDA_{PL}$ (ours)	**61.9**	**54.8**	**65.6**	**50.3**	**49.7**	52.6	**46.8**	48.1	**50.6**

3. CosMix[9]



The CoSMix model can be broadly categorized into three primary components: Semantic Selection, Compositional Mix, and Training and Update, which aim at facilitate robust and adaptive point cloud segmentation across domains. Each section plays a critical role in enhancing the model's capability to generalize and perform effectively in diverse and challenging scenarios.This paragraph provides a comprehensive analysis of these three sections, elaborating on their respective roles and the methodologies employed within each.

3.1 Semantic Selection

The first stage, Semantic Selection, is crucial for identifying and extracting informative and reliable point cloud patches from the source and target domains. This process leverages class frequency distributions to dynamically sample classes, ensuring that the selection is both representative and balanced. The approach adapts to the data distribution at each iteration, enhancing the model's ability to handle long-tailed classes. Specifically, for the source point cloud, we compute the class frequency distribution and utilize a weighted random sampling function to select a subset of classes. The likelihood of selecting a class is inversely proportional to its frequency, promoting the inclusion of underrepresented classes. Additionally, in the experiments,

the number of patches selected from available classes is assigned based on prior experience, and the value is set by examining the long-tail classes present in the source domain during the adaptation process.

In the context of Semi-Supervised Domain Adaptation (SSDA), the patch selection methodology mirrors that employed for the source domain, wherein different classes are selected based on known labels. Conversely, in Unsupervised Domain Adaptation (UDA), the target domain data initially lacks corresponding labels. However, through the learning process of the teacher network, pseudo labels are generated, facilitating the subsequent steps in the adaptation process.

3.2 Compositional Mix

Compositional Mix module combines and mixes the features extracted from both source domain and target domain. Considering of the spatial consistency and semantic information, the hybrid point cloud data generated in the process can retain the key features of the original data. The Compositional Mix process is carried out on two branches separately. It involves three consecutive steps: local random augmentation, concatenation, and global random augmentation.

Local random enhancement is applied to the patches obtained in the semantic selection phase to increase data variability. These augmented patches are then concatenated with point clouds from the other domain, forming mixed point clouds. Finally, the mixed point clouds are subjected to global random augmentation to further enhance their diversity.

3.3 Training and Update

From the first two steps, both branches obtain mixed point cloud data. In the training stage, during each batch iteration, the student network parameters are updated to minimize the total objective loss, which is the sum of the losses of the two branches. The segmentation loss is computed for the mixed point clouds, targeting the reduction of segmentation errors in both domain transfers. The loss of a single branch is calculated using Dice segmentation loss, known for its effectiveness in handling large-scale point clouds with long-tailed class distributions.

In the experiment, the learning update of the student network is only performed in a single branch, and its parameters will be shared with the student network of another branch after completion. Then, the parameters of the teacher network are updated via the exponential moving average (EMA) method： $\theta_i^{'} = \beta\theta_{i-1}^{'} + (1-\beta)\theta$, ensuring stable pseudo-label generation and maintaining high average confidence.

四．不理解的点

1．数据增强的实现

在 CoSMix 模型中的 Compositional Mix 阶段中，有两个操作，分别是局部随机增强和全局随机增强，我们知道数据增强的目的是通过数据变换帮助模型更好的学习局部特征，增强在不同全局条件下的鲁棒性和提高模型的泛化能力和性能，但是在具体实验过程中，这个过程通过很少的代码在 get_data 中就完成了，我们没有找到办法探知具体进行增强过程和效果。

2.

五．可能的改进

CosMix 模型的改进:

1. 在数据增强技术中按照合适的方法增加颜色扰动，提高模型适应不同光照条件下的点云，增强其鲁棒性；在局部增强和全局增强中可以通过放缩和不同轴的旋转，增加模型对物体不同大小和视角的适应性。

2. 在生成为标签时，可以设计多个教师网络，在不同训练回合进行更新，通过更新加权系数，利用集成学习的方法，提高伪标签的准确性和稳定性。

3. 模型在实现过程中对于不同的数据集采用了不同的超参数，如置信概率等。可以通过调整这些超参数进一步探索模型。

AUDA 的改进思路:

1.在分割头前引入特征融合（晚融合），把 2D 和 3D 特征拼接在一起，以提高模态间的鲁棒性。

2.除了考虑 AUDA，对于一般的点云分割 UDA 任务，我猜测，也许设计一种点云数据标注器，让整个模型在半合成数据集上训练可能会得到不错的效果。一开始用全人工标注，后续逐渐增加生成数据的比例。

六．收获与总结

侯雪冰：通过阅读论文和学习代码，对点云语义分割及迁移学习有了进一步的理解：基础知识，发展历程和不同的点云语义分割的方法。同时，我更加了解如何阅读一篇论文中的重点部分，结合框图与文献深入了解方法算法的实现细节，对每一部分原理和目的深入理解也让我在论文观点的基础上有自己的思考。

韩骐骏：第一，我理解了对于深度学习，数据其实是比模型更重要的，或者说，其实模型只是把数据利用到什么程度罢了。那么如何利用数据呢？通过本次项目的学习，我认为应该充分利用“信息”。假如没有充分利用信息，那么即使数据集很大，训练出的结果也不一定好。第二，在做一项研究时，明确课题后，应该找一篇该领域经典的论文，把他的思路理解透彻，明白这篇论文的核心观点及其原因，再根据和这篇文章相关的关键词检索论文数据库，看看后人是怎么评价和改进这个模型的。

欧阳莹轩：在学习新领域的过程中对深度学习、迁移学习和语义分割方面的知识有了初步的了解，对于其中用到了算法、公式和技术进行了探讨学习。复现代码的过程中增加了编程的经验，也在学习文献的过程中收获了文献阅读的思路，会抓住重点和原理图去理解。

七. 参考文献

稀疏卷积 Sparse Convolution Net-CSDN 博客
多模态学习(MultiModal Learning) - 张浩在路上 (imzhanghao.com)
我们真的需要那么多标注数据吗？半监督学习技术近年来的发展历程及典型算法框架的演进 - 知乎 (zhihu.com)
对抗训练浅谈：意义、方法和思考（附 Keras 实现）—CSDN 博客
[4]Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In CVPR, 2018.
[5]Jaritz, M., Vu, T.H., Charette, R.d., Wirbel, E., P ´ erez, P., 2020. xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 12605 – 12614.
[6]Maximilian Jaritz1, *Jiayuan Gu† , Hao Su† Inria/Valeo* , UC San Diego†, Multi-view PointNet for 3D Scene Understanding, in CVPR, 2019
[7] Wei Liu a , Zhiming Luo b,* , Yuanzheng Cai c , Ying Yu a , Yang Ke d , Jos´e Marcato Junior e , Wesley Nunes Gonçalves e , Jonathan Li d. Adversarial unsupervised domain adaptation for 3D semantic segmentation with multi-modal learning, ISPRS Journal of Photogrammetry and Remote Sensing, 2021, P211-221,
[8]Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, Patrick Perez.ADVENT: Adversarial Entropy Minimization for Domain Adaptation in Semantic Segmentation, 2017
[9]Cristiano Saltori, Fabio Galasso, Giuseppe Fiameni, Nicu Sebe, Fabio Poiesi, and Elisa Ricc. Compositional Semantic Mix for Domain Adaptation in Point Cloud
Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015