

重新审视半监督语义分割中强弱一致性

摘要

在这项工作中，作者重新审视了由半监督分类网络 FixMatch 推广的强弱一致性框架，其中对弱扰动图像的预测可作为对其强扰动视图的监督。它的成功在很大程度上依赖于人工设计的强数据增强。然而，这可能是有限的并且不足以探索更广泛的扰动空间。因此，作者提出了一个扩展的扰动空间以及一种双流扰动技术来提升网络的性能。但作者未考虑到固定的高阈值会使得模型在早期训练阶段的数据利用率较低以及伪标签生成过程中的类别不平衡问题。为此考虑使用自适应阈值和自适应类别公平来解决上述问题，实验结果表明改进的方法优于原文的方法。

关键词：语义分割 自适应 强弱一致性 半监督

1 引言

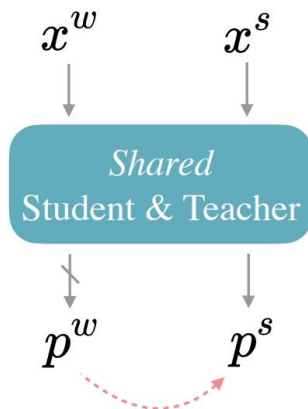


图 1. Fixmatch 的核心思想图

语义分割旨在为图像提供像素级预测，这可以被视为密集分类任务，并且是自动驾驶等现实应用的基础。然而，传统的完全监督场景非常需要人类精确标记的图像，这极大地阻碍了其在某些领域的广泛应用，尤其是在某些领域中标注大量图像成本极其昂贵，甚至不可行。因此，半监督语义分割被提出并受到越来越多的关注。

半监督语义分割的先进方法已经从基于 GAN 的对抗训练范式[1]发展到广泛采用的一致性正则化框架[4, 5]和自监督学习范式[6]。在这项工作中，我们重点关注 FixMatch[7]在

半监督分类领域推广的弱到强一致性正则化框架。其核心思想是用强扰动分支产生的预测 x^s 来监督弱扰动分支产生的预测 x^w ，如图 1 所示。直观地说，它的成功在于模型更有可能对 x^w 产生高质量的预测，而 x^s 对我们的模型学习更有效，因为强扰动引入了额外的信息并减轻了确认偏差。

表 1. 图像级强扰动（SP）对 Pascal 数据集上 FixMatch 的重要性。w/o any SP:表示不使用任何强扰动；w/ CutMix::表示只使用 CutMix 扰动；w/ whole SP: 表示使用颜色变换以及 CutMix

Method	# labeled images (10582 in total)				
	92	183	366	732	1464
w/o any SP	39.5	52.7	65.5	69.2	74.6
w/ CutMix	<u>56.7</u>	<u>67.9</u>	<u>71.9</u>	<u>75.1</u>	<u>78.3</u>
w/ whole SP	63.9	73.0	75.5	77.8	79.2

如表 1 所示，是否采用扰动之间的性能差距极其巨大。受到这些线索的启发，作者同时从两个不同的视角和方向进一步强化强扰动，即拓展更广阔的扰动空间，充分收获原始扰动。图像级扰动，例如，颜色抖动和 CutMix[8]包括启发式偏差，其实际上将额外的先验信息引入到 FixMatch 范式中，以便捕获一致性正则化的优点。如果没有配备这些扰动，FixMatch 将退化为一个简单的在线自我训练，产生更糟糕的结果。为了扩展原始扰动空间，我们为原始图像和提取的特征设计了统一的扰动框架。另一方面，当前的 FixMatch 框架仅利用小批量中每个未标记图像的单个强视图，这不足以充分利用预定义的扰动空间。考虑到这一点，作者对输入提出了一种简单又高效的改进方法，从扰动池中随机采样两个独立的强视图。然后，他们被并行输入学生模型，并同时受到他们共享的弱视图的监督。直观上，强制两个强视图接近一个共同的弱视图可以被视为最小化这些强视图之间的距离。因此，它具有对比学习的优点，可以学习更多判别性表示，并被证明对我们当前的任务特别有益。

2 相关工作

2.1 半监督学习

半监督学习的核心问题在于如何针对无标签数据设计合理有效的监管信号。两个主要的方法被提出来解决这个问题，即熵最小化[9–11]和一致性正则化[12, 13]。熵最小化为未标记的数据分配伪标签，然后将它们与人工标记的数据相结合以进行进一步的训练。一致

性正则化假设未标记样本的预测对于不同形式的扰动应该是不变的。其中，FixMatch[7]提出向未标记图像注入强扰动，并使用弱扰动图像的预测来监督训练过程，以实现这两种方法的优点。然而，FlexMatch[14]考虑不同类别的学习状态，使用类别阈值过滤低置信度标签。

2.2 半监督语义分割

先前的工作将 GAN[15]作为未标记图像的辅助监督，用于区分伪标签和手工标签。受半监督学习快速发展的推动，最近的方法[16–18]从一致性正则化和熵最小化的角度努力实现更简单的训练范式。在这种趋势下，French 等人[19]发现 Cutout 和 CutMix 是一致性正则化在分割中取得成功的关键。随后，AEL [20]设计了一个自适应的 CutMix 和采样策略，以加强对表现不佳类别的学习。受对比学习的启发，Lai 等人[21]提出在不同的上下文语义下强制执行共享块的预测。U2PL[22]将不确定的像素视为可靠的负样本，以与相应的正样本进行对比。CPS[23]引入了双重模型来相互监督，类似于共同训练的核心思想。

熵最小化研究领域的其他工作利用自训练以离线方式为未标记图像分配伪标签。从这个角度来看，袁等人[24]声称对未标记图像的过度扰动对于干净数据分布来说是灾难性的，因此提出对这些图像进行单独的批量归一化。同时，ST++[25]指出，适当的强数据扰动确实对自训练非常有帮助。此外，为了解决伪标签中遇到的类别偏差问题，He 等人[26]对齐手工标签和伪标签之间的类分布。USRN[27]将平衡子类分布聚类作为正则化，以减轻预定义类的不平衡。

3 本文方法

3.1 图像和特征的统一扰动

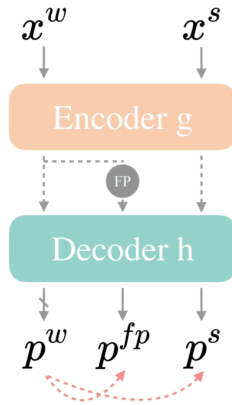


图 2. UniPerb 网络结构

尽管 Fixmatch 很受欢迎，但它的性能在很大程度上取决于研究人员精心设计的强扰动，

其最佳组合和超参数是耗时的。此外，在某些情况下，如医学图像分析和遥感图像解释，它可能需要特定领域的知识找出有前途的方法。更重要的是，它们完全受限于图像级别，阻碍了学生模型针对更多样化的扰动保持多级一致性。

为了构建一个更广泛的扰动空间，本文在弱扰动图像 x^w 的特征上注入扰动，称为特征扰动。本文选择将不同级别的扰动分离为多个独立的前馈流，使学生模型能够更直接地在每个流中实现目标一致性。形式上，分割模型 f 可以被分解为编码器 g 和解码器 h 。除了在 FixMatch 中获得 p^w 和 p^s 之外，还通过以下方式从辅助特征扰动流中获得 p^{fp} ：

$$e^w = g(x^w), \quad (1)$$

$$p^{fp} = h(\mathcal{P}(e^w)), \quad (2)$$

其中 e^w 是 x^w 的提取特征， \mathcal{P} 表示特征扰动，例如，dropout 或添加均匀噪声。

如图 2 所示，为每个未标记的小批量维护三个前馈流，它们是 (i) 最简单的流： $x^w \rightarrow f \rightarrow p^w$ ，(ii) 图像级强扰动流： $x^s \rightarrow f \rightarrow p^s$ ，以及 (iii) 本文引入的特征扰动流： $x^w \rightarrow g \rightarrow \mathcal{P} \rightarrow h \rightarrow p^{fp}$ 。通过这种方式，学生模型被强制要求在图像和特征级别上与统一扰动一致，本文将其命名为 UniPerb。无监督损失 \mathcal{L}_u 被公式化为：

$$\mathcal{L}_u = \frac{1}{B_u} \sum \mathbb{I}(\max(p^w) \geq \tau) (H(p^w, p^s) + H(p^w, p^{fp})), \quad (3)$$

3.2 双流扰动

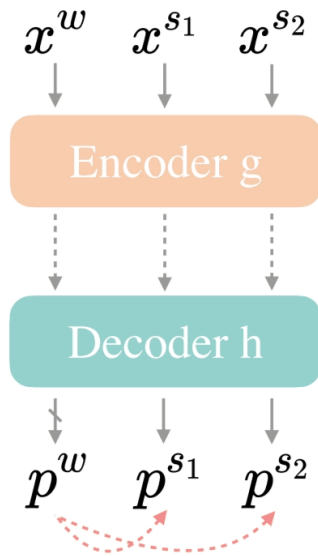


图 3. 图 3. DusPerb 网络结构

受到图像级强扰动的巨大优势的激励，作者希望充分探索它们。作者受到自监督学习和半监督分类方面最新进展的启发，为未标记数据构建多个视图作为输入可以更好地利用扰动。例如，SwAV[28]提出了一种名为“多视图”的新技术，可以在不同分辨率的视图实现从局部到全局的一致性。同样，ReMixMatch[29]生成多个强增强视图供模型学习。

因此，作者通过强扰动池 \mathcal{A}^s 独立地从 x^w 产生双流扰动 (x^{s1}, x^{s2}) 。由于 \mathcal{A}^s 是预定义的且具有不确定性，因此 x^{s1} 和 x^{s2} 不相等。该双流扰动框架 (DusPerb) 如图 3 所示。

作者通过消融研究证实，这种微小的改变比原始的 Fixmatch 带来了实质性的变化，并取得了最好的结果。用共享的弱增强视图对两个强增强视图进行正则化，也可以被视为强制这两个强视图之间保持一致。假设 k_w 是 x^w 预测的类别的分类器权重，而 (q_{s1}, q_{s2}) 是图像 (x^{s1}, x^{s2}) 的特征，那么在我们采用的交叉熵损失中，我们针对 $\sum_{i=0}^C q_j \cdot k_i$ 最大化 $q_j \cdot k_w$ ，其中 $j \in \{s1, s2\}$ ， k_i 是类别 i 的分类器权重。因此可以认为我们也在最大化 q_{s1} 和 q_{s2} 之间的相似性。因此满足 InfoNCE 损失：

$$\mathcal{L}_{s1 \leftrightarrow s2} = -\log \frac{\exp(q_{s1} \cdot q_{s2})}{\sum_{i=0}^C \exp(q_j \cdot k_i)}, \quad s.t., j \in \{s1, s2\}, \quad (4)$$

其中， q_{s1} 和 q_{s2} 是正样本对，而除 k_w 之外的所有其他分类器权重都是负样本。

3.3 整体网络框架

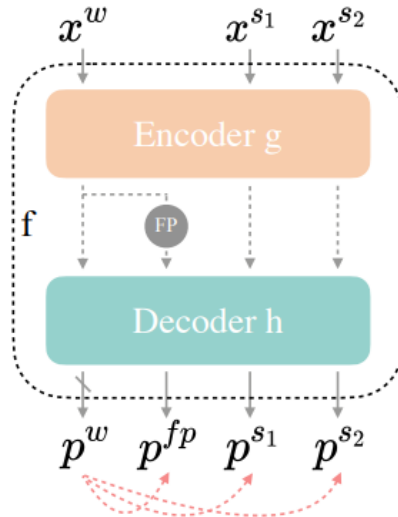


图 4. UniMatch 网络框架

总而言之，作者提出了两种利用未标记图像的关键技术，即 UniPerb 和 DusPerb。作者集成了这两种方法的整体框架（称为 UniMatch）如图 4 所示。与 FixMatch 相比，增加了两个辅助前馈流，一个用于 x^w 特征的扰动，另一个用于 (x^{s1}, x^{s2}) 的多视图学习。最终

的无监督损失为：

$$\mathcal{L}_u = \frac{1}{B_u} \sum \mathbb{I}(\max(p^w) \geq \tau) \cdot \left(\lambda H(p^w, p^{fp}) + \frac{\mu}{2} (H(p^w, p^{s1}) + H(p^w, p^{s2})) \right) \quad (5)$$

4 复现细节

4.1 创新点

4.1.1 改进一：自适应阈值

半监督学习的关键是阈值的设定，因此将原论文中的固定阈值改为自适应阈值。自适应阈值在训练过程中利用模型预测来自动定义和自适应调整每个类别的置信度阈值。自适应阈值首先估计一个全局阈值作为模型置信度的指数移动平均。然后，通过每个类别的局部阈值来调整全局阈值。当训练开始时，阈值较低，可以接受更多可能正确的样本进行训练。随着模型变得更加自信，阈值会自适应地增加，以过滤掉可能不正确的样本，从而减少确认偏差。

(1) 自适应全局阈值

全局阈值应该与模型对未标记数据的置信度相关，反映整体学习状态。此外，全局阈值应在训练期间稳定增加，以确保丢弃不正确的伪标签。将全局阈值 τ_t 设置为模型对未标记数据的平均置信度，其中 t 表示第 t 个时间步长（迭代）。将全局置信度估计为每个时间步置信度的指数移动平均值。我们将 τ_t 初始化为 $\frac{1}{C}$ ，其中 C 表示类的数量。全局阈值定义为：

$$\tau_t = \begin{cases} \frac{1}{C}, & \text{if } t=0, \\ \lambda \tau_{t-1} + (1-\lambda) \frac{1}{\mu B} \sum_{b=1}^{\mu B} \max(q_b), & \text{otherwise,} \end{cases} \quad (6)$$

其中 $\lambda \in (0,1)$ 是指数移动平均的衰减系数， q_b 表示弱增强视图， μ 是无标签数据与标签数据的比值。

(2) 自适应局部阈值

局部阈值旨在以类特定的方式调整全局阈值，以考虑类内多样性和可能的类邻接。我们计算模型对每个类 C 的预测的期望值，以估计特定于类的学习状态：

$$\tilde{p}_t(c) = \begin{cases} \frac{1}{C}, & \text{if } t=0, \\ \lambda \tilde{p}_{t-1}(c) + (1-\lambda) \frac{1}{\mu B} \sum_{b=1}^{\mu B} q_b(c), & \text{otherwise,} \end{cases} \quad (7)$$

其中 $\tilde{p}_t = [\tilde{p}_t(1), \tilde{p}_t(2), \dots, \tilde{p}_t(C)]$ 是包含所有 $\tilde{p}_t(C)$ 的列表，将全局阈值和局部

阈值相结合，我们得到最终的自适应阈值 $\tau_t(C)$ 为：

$$\tau_t(c) = \text{MaxNorm}(\tilde{p}_t(c)) \cdot \tau_t = \frac{\tilde{p}_t(c)}{\max\{\tilde{p}_t(c) : c \in [C]\}} \cdot \tau_t, \quad (8)$$

其中 **MaxNorm** 是最大归一化，第 t 次迭代的无监督损失函数为：

$$\mathcal{L}_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{I}(\max(q_b) > \tau_t(\arg \max(q_b))) \cdot \mathcal{H}(\hat{q}_b, Q_b) \quad (9)$$

其中 $\mathcal{H}(\cdot, \cdot)$ 表示交叉熵损失， $\mathbb{I}(\cdot > \tau)$ 是以 τ 为阈值的指示函数。

4.1.2 改进二：自适应类公平

类公平性鼓励模型对每个类进行不同的预测，从而产生有意义的自适应阈值，特别是在标签数据很少的情况下。利用公式 7 计算的 \tilde{p}_t 作为无标签数据的预测分布期望估计并优化了 \tilde{p}_t 和 $\bar{p} = \mathbb{E}_{\mu B} [p_m(y | \Omega(u_b))]$ 在小批量上的交叉熵作为 $H(\mathbb{E}_u [p_m(y | u)])$ 的估计。考虑到潜在的伪标签分布可能不一致，提出了一种自适应调整公平性目标的方法，即通过伪标签的直方图分布规范化概率期望，以抵消不平衡的负面影响：

$$\bar{p} = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{I}(\max(q_b) \geq \tau_t(\arg \max(q_b))) Q_b \quad (10)$$

$$\bar{h} = \text{Hist}_{\mu B}(\mathbb{I}(\max(q_b) \geq \tau_t(\arg \max(q_b))) \hat{Q}_b) \quad (11)$$

$$\tilde{h}_t = \lambda \tilde{h}_{t-1} + (1-\lambda) \text{Hist}_{\mu B}(\hat{q}_b) \quad (12)$$

其中 Q_b 表示强增强视图， \hat{q}_b 是由 q_b 转换而来的 one-hot 标签。

第 t 次迭代的自适应类公平损失函数为：

$$\mathcal{L}_f = -\mathcal{H}\left(\text{SumNorm}\left(\frac{\tilde{p}_t}{\tilde{h}_t}\right), \text{SumNorm}\left(\frac{\bar{p}}{\bar{h}}\right)\right) \quad (13)$$

其中 $\text{SumNorm} = (\cdot) / \sum(\cdot)$ 。通过直方图分布归一化，自适应类公平鼓励每个小批量

的输出概率的期望值接近模型的边际类分布。这有助于模型产生多样化的预测，尤其是在几乎没有监督的情况下，从而收敛更快以及泛化更好。

4.2 与已有开源代码对比

原文方法的作者已开源代码，在此基础上增加自己创新点的代码，具体新增模块代码如下：

(1) 伪标签生成

代码：

```
import torch
def smooth_targets(logits, targets, smoothing=0.1):
    """
    标签平滑
    """
    with torch.no_grad():
        true_dist = torch.zeros_like(logits)
        true_dist.fill_(smoothing / (logits.shape[-1] - 1))
        true_dist.scatter_(1, targets.data.unsqueeze(1), (1 -
smoothing))
    return true_dist

class PseudoLabeling:
    """
    伪标签生成
    """
    def __init__(self):
        super().__init__()

    @torch.no_grad()
    def gen_ulb_targets(self,
                        logits,
                        use_hard_label=True,
                        T=1.0,
                        softmax=False,
                        label_smoothing=0.0):

        logits = logits.detach()
        if use_hard_label:
            pseudo_label = torch.argmax(logits, dim=1)
            if label_smoothing:
                pseudo_label = smooth_targets(logits, pseudo_label,
label_smoothing)
```



```

        return pseudo_label

    # 返回软标签
    if softmax:
        pseudo_label = torch.softmax(logits / T, dim=-1)
    else:
        pseudo_label = logits
    return pseudo_label

```

代码说明：

`gen_ulb_targets()` 函数根据传入的逻辑值或者概率值生成伪标签。`logits` 表示输入；`use_hard_label` 表示是否使用硬标签，为真表示使用硬标签否则使用软标签；`softmax` 表示输入数据是否要进行 softmax 操作，当 `softmax` 为真表示要进行 softmax 操作否则不用进行 softmax 操作；`T` 表示温度系数；`label_smoothing` 表示标签平滑因子。

输入数据首先判断使用硬标签还是软标签，然后再判断是否使用 softmax 操作，最后进行标签平滑操作。

(2) 自适应阈值与自适应类公平

代码：

```

class FreeMatchThresholding:
    """
    自适应阈值与自适应类公平
    """
    def __init__(self, num_classes, momentum=0.999, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.num_classes = num_classes
        self.m = momentum
        self.p_model = torch.ones((self.num_classes)) /
self.num_classes # 自适应局部阈值
        self.label_hist = torch.ones((self.num_classes)) /
self.num_classes # 标签直方图
        self.time_p = self.p_model.mean() # 自适应全局阈值

    @torch.no_grad()
    def update(self, probs_x_ulb):
        if algorithm.distributed and algorithm.world_size > 1:
            probs_x_ulb = concat_all_gather(probs_x_ulb)
            max_probs, max_idx = torch.max(probs_x_ulb,
dim=1, keepdim=True)

            if algorithm.use_quantile:

```

```

        self.time_p = self.time_p * self.m + (1 - self.m) *
torch.quantile(max_probs,0.8) #* max_probs.mean()
    else:
        self.time_p = self.time_p * self.m + (1 - self.m) *
max_probs.mean()

    if algorithm.clip_thresh:
        self.time_p = torch.clip(self.time_p, 0.0, 0.95)

    self.p_model = self.p_model * self.m + (1 - self.m) *
probs_x_ulb.mean(dim=[0,2,3])
    hist = torch.bincount(max_idx.reshape(-1),
minlength=self.p_model.shape[0]).to(self.p_model.dtype)

    print('hist:',hist.shape)
    self.label_hist = self.label_hist * self.m + (1 - self.m) *
(hist / hist.sum())

    @torch.no_grad()
    def masking(self,logits_x_ulb, softmax_x_ulb=True, *args,
**kwargs):
        if not self.p_model.is_cuda:
            self.p_model = self.p_model.to(logits_x_ulb.device)
        if not self.label_hist.is_cuda:
            self.label_hist = self.label_hist.to(logits_x_ulb.device)
        if not self.time_p.is_cuda:
            self.time_p = self.time_p.to(logits_x_ulb.device)

        if softmax_x_ulb:
            probs_x_ulb = torch.softmax(logits_x_ulb.detach(), dim=-1)
        else:
            # logits is already probs
            probs_x_ulb = logits_x_ulb.detach()

        self.update(probs_x_ulb)

        max_probs, max_idx = probs_x_ulb.max(dim=-1)
        mod = self.p_model / torch.max(self.p_model, dim=-1)[0]
        mask = max_probs.ge(self.time_p *
mod[max_idx]).to(max_probs.dtype)
        return mask, self.p_model, self.label_hist

```

代码解释：

num_classes 表示数据的类别数量；m 表示指数移动平均的加权权重值；p_model 表示

自适应局部阈值；time_p 表示自适应全局阈值；label_hist 表示标签直方图。

masking() 函数用来计算并返回 mask、p_model 和 label_hist。

输入数据首先判断是否放在 GPU 上，没有则将数据放在 GPU 上。然后判断是否使用 softmax 操作，接着调用 update() 函数用于计算并更新 p_model、time_p 和 label_hist 参数（即根据公式 6-8 和公式 10-12），最后通过相关操作得到 mask 并返回计算结果。

4.3 实验设置

主要采用基于 ResNet 的 DeepLabv3+作为我们的分割模型。ResNet 在所有实验中使用 16 的输出步长，以节省内存并加快训练速度。在训练过程中，每个小批量由 8 张标记图像和 8 张未标记图像组成。使用 SGD 优化器，Pascal 的初始学习率设置为 0.001。该模型在多学习率调度器下训练了 80 个 epoch。将 CutMix 和颜色抖动组合起来形成扰动池 As。将原始图像调整为 0.5 到 2.0 之间的大小、裁剪并翻转以获得其弱增强版本 x^w 。数据集的训练分辨率设置为 321，采用 50%概率的通道丢失作为特征扰动，将其插入编码器和解码器中。

5 实验结果分析

表 2 在 pascal 数据集上复现实验结果比较。1/8 和 1/4 表示标签数据占的比例，括号中的数字表示标签数据的大小

方法	1/8 (183)	1/4 (366)
原文方法	77.2	78.8
改进方法	78.4	82.3

由实验结果可知，创新点自适应阈值和自适应类公平提高了原文的性能，并且在标签数据更多的情况下表现越好。取得提升的效果来源于，原文使用固定的高阈值以保证伪标签的质量，然而固定的高阈值可能会导致早期训练阶段的数据利用率较低。自适应阈值在训练过程中动态调节阈值的大小以充分利用无标签数据，自适应类公平解决了伪标签生成过程中的类别不平衡问题，极大提升了模型对每个类别预测的准确性。

6 总结与展望

作者研究了 FixMatch 在半监督语义分割中的作用，发现适当的图像级强扰使得模型确实可以取得 SOTA 结果。受此启发，作者从两个角度进一步拓宽扰动空间。一方面，统一图像和特征级的扰动，形成更加多样化的扰动空间。另一方面，我们设计了一种双流扰动技术来充分利用图像级扰动。最终使得 UniMatch 在自然、医学和遥感场景中显著提升了性

能表现。

但作者使用的固定高阈值不能充分利用无标签数据，导致模型在早期训练阶段效果比较差，因此考虑使用自适应阈值来动态调节阈值，以达到充分利用无标签的数据。原文作者也没考虑到伪标签生成过程中的类别不平衡问题，因此考虑使用自适应类别公平，提升模型对每个类别的预测准确性。

未来可以进一步考虑是否可以将此技术引入跨域语义分割以及结合当下火热的大模型技术结合起来做研究。

参考文献

- [1] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In NeurIPS, 2014.
- [2] Sudhanshu Mittal, Maxim Tatarchenko, and Thomas Brox. Semi-supervised semantic segmentation with high-and low-level consistency. TPAMI, 2019.
- [3] Nasim Souly, Concetto Spampinato, and Mubarak Shah. Semisupervised semantic segmentation using generative adversarial network. In ICCV, 2017.
- [4] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In CVPR, 2020.
- [5] Yuliang Zou, Zizhao Zhang, Han Zhang, Chun-Liang Li, Xiao Bian, Jia-Bin Huang, and Tomas Pfister. Pseudoseg: Designing pseudo labels for semantic segmentation. In ICLR, 2021.
- [6] Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. St++: Make self-training work better for semi-supervised semantic segmentation. In CVPR, 2022.
- [7] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In NeurIPS, 2020.
- [8] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In ICCV, 2019.
- [9] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In NeurIPS, 2005.
- [10] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In CVPR, 2021.
- [11] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In CVPR, 2020.
- [12] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In ICLR, 2020.
- [13] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In NeurIPS, 2019.
- [14] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and

- Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In NeurIPS, 2021.
- [15] Inigo Alonso, Alberto Sabater, David Ferstl, Luis Montesano, and Ana C Murillo. Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank. In ICCV, 2021.
 - [16] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In NeurIPS, 2014.
 - [17] Zhengyang Feng, Qianyu Zhou, Qiqi Gu, Xin Tan, Guangliang Cheng, Xuequan Lu, Jianping Shi, and Lizhuang Ma. Dmt: Dynamic mutual training for semi-supervised learning. PR, 2022.
 - [18] Zhanghan Ke, Di Qiu, Kaican Li, Qiong Yan, and Rynson WH Lau. Guided collaborative training for pixel-wise semi-supervised learning. In ECCV, 2020.
 - [19] Geoff French, Timo Aila, Samuli Laine, Michal Mackiewicz, and Graham Finlayson. Semi-supervised semantic segmentation needs strong, high-dimensional perturbations. In BMVC, 2020.
 - [20] Hanzhe Hu, Fangyun Wei, Han Hu, Qiwei Ye, Jinshi Cui, and Liwei Wang. Semi-supervised semantic segmentation via adaptive equalization learning. In NeurIPS, 2021.
 - [21] Xin Lai, Zhuotao Tian, Li Jiang, Shu Liu, Hengshuang Zhao, Liwei Wang, and Jiaya Jia. Semi-supervised semantic segmentation with directional context-aware consistency. In CVPR, 2021.
 - [22] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-supervised semantic segmentation using unreliable pseudo-labels. In CVPR, 2022.
 - [23] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In CVPR, 2021.
 - [24] Jianlong Yuan, Yifan Liu, Chunhua Shen, Zhibin Wang, and Hao Li. A simple baseline for semi-supervised semantic segmentation with strong data augmentation. In ICCV, 2021.
 - [25] Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. St++: Make self-training work better for semi-supervised semantic segmentation. In CVPR, 2022.
 - [26] Ruifei He, Jihan Yang, and Xiaojuan Qi. Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation. In ICCV, 2021.
 - [27] Dayan Guan, Jiaxing Huang, Aoran Xiao, and Shijian Lu. Unbiased subclass regularization for semi-supervised semantic segmentation. In CVPR, 2022.
 - [28] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In NeurIPS, 2020.
 - [29] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In ICLR, 2020.