



Randomized Spectrum Transformations for Adapting Object Detector in Unseen Domains

汇报人：吴沁哲 刘晓

指导教师：张磊





目录

Contents

01 研究背景与相关工作

02 研究内容与方法

03 实验结果与分析

04 总结

1

研究背景与相关工作

1.1 研究背景

目标检测 (object detection)

旨在定位和识别图像中已知类别实例的问题，是计算机视觉领域的核心问题。

目标检测面临的挑战

域偏移 (domain shift) 问题。当前的目标检测基于**独立同分布**假设。该假设在现实中很容易被打破。检测器**性能会急剧下降**。

解决方法

领域适应 (Domain Adaptation, 简称DA)

领域泛化 (Domain Generalization, 简称DG)



图1 检测器在正常天气和大雾天下检测的效果对比^[1]

[1] He Z, Zhang L, Gao X, et al. Multi-adversarial faster-RCNN with paradigm teacher for unrestricted object detection[J]. International Journal of Computer Vision, 2023, 131(3): 680-700.

1.1 研究背景

领域适应 DA

在训练时可以使用**目标领域的未标注数据**，通过**对源领域和目标领域的数据进行对齐**，从而提高模型在目标领域的性能。

但是，在实际环境中，从**所有**可能的目标域中收集数据是不切实际的。

领域泛化 DG

在训练时**完全不使用目标领域的数据**，只依赖多个源领域的数据，希望训练出的模型能够直接在未见过的目标领域上表现良好。

1. 多个多样性的源域
2. 学习领域不变特征
3. 应用于未见过的目标域

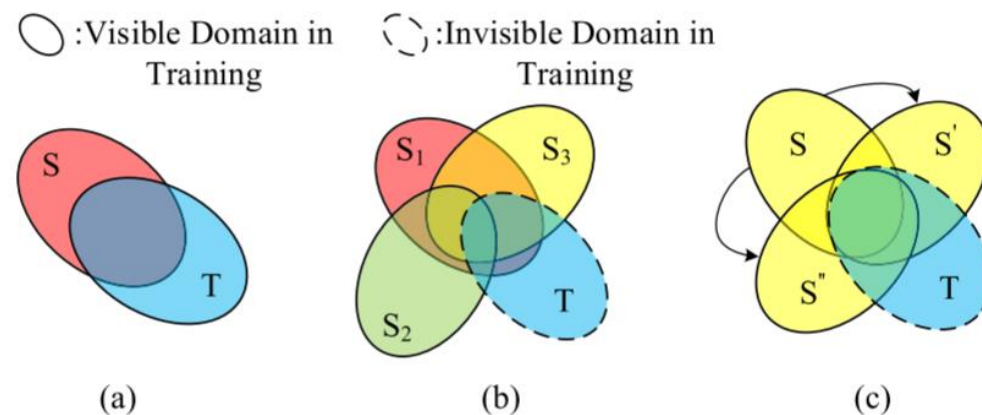


图2 不同的泛化问题:(a)领域适应将标记的源(S)和未标记的目标(T)域之间的数据分布对齐;(b) 领域泛化(DG)学习多源域的不变性;(c)单领域泛化(SDG)在单个域上学习增广源的不变性。

DG常用于图像分类中的泛化问题，本文将探索，目标检测的泛化问题，即称为领域泛化目标检测问题((domain generizable object detection , DGOD)

1.1 研究背景

领域泛化所遇到的问题

源域多样性问题

领域泛化方法需要足够多样化的源域来帮助学习领域不变信息。现有的DG方法容易受到源域性质的影响，容易与源域过拟合。这大大降低了DG方法的泛化性能。

我们将这个问题称为**域偏差问题(domain bias)**。同时，我们观察到梯度之间的不平衡可以用来反应域偏差。

本文提出的解决方法

随机频谱变换(Randomized Spectrum Transformation)通过数据增强方法增加源域的多样性
梯度加权(Gradient Weighting) 从梯度平衡的角度自然地解决域偏差

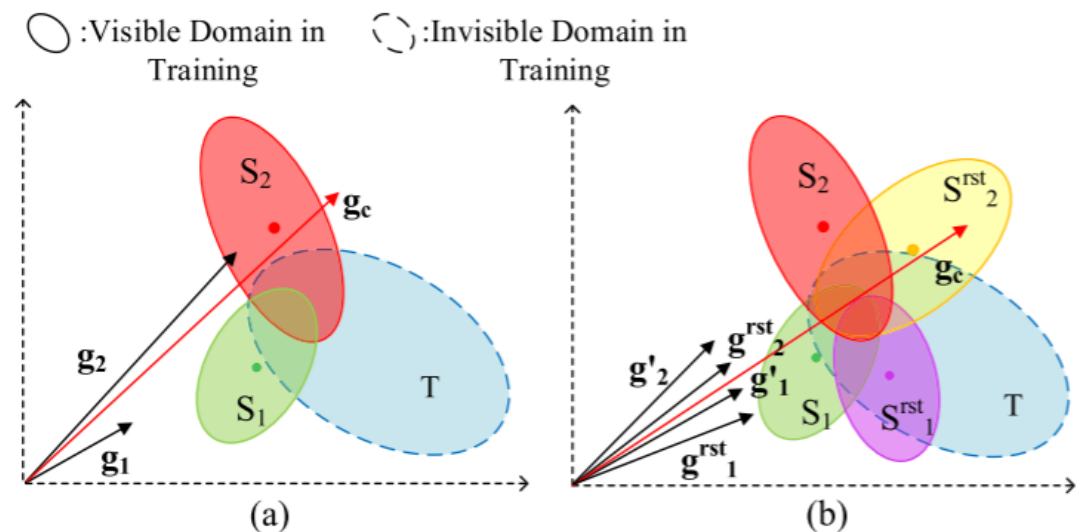


图3 域偏差问题图解

1.2 相关工作

领域适应目标检测相关研究

基于对抗/重建的方法

这两种方法的核心目标是减少源域和目标域之间的分布差异，从而提高目标检测模型在目标域上的性能。

基于对抗的方法

基于对抗的方法通过对抗性训练来对齐源域和目标域的特征分布，从而提高目标检测模型在目标域上的性能。

■ Chen等人将对抗性学习应用于DAOD领域^[1]

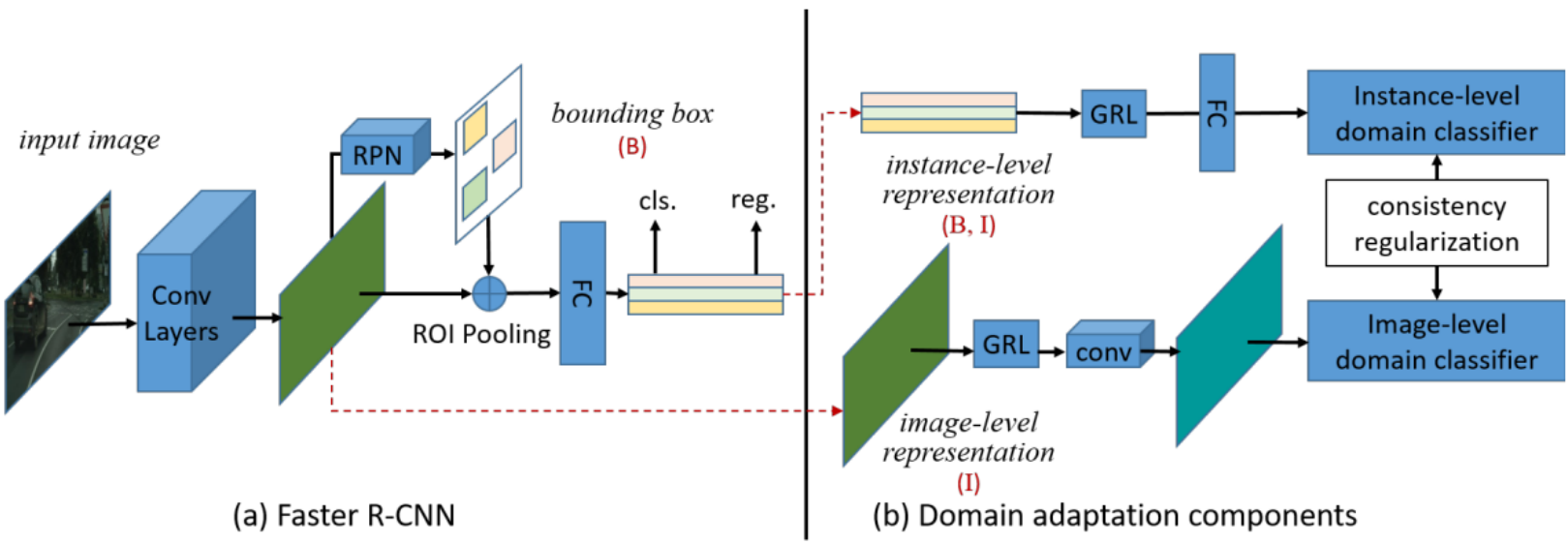


图4 Chen文中的领域适应Faster R-CNN模型总览^[1]

- **特征提取器**
从源域和目标域的数据中提取特征
- **域分类器**
尝试区分特征是来自源域还是目标域
- **对抗训练**
特征提取器和域分类器进行对抗训练。

[1] Chen Y, Li W, Sakaridis C, et al. Domain adaptive faster r-cnn for object detection in the wild[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 3339-3348.

1.2 相关工作

领域适应目标检测相关研究

基于重建的方法

基于重建的方法通过**重构**目标域的数据，**使得源域和目标域的特征更加一致**。这种方法通常涉及自编码器或生成对抗网络（GANs），以生成目标域数据的重构表示。

- Vinicius F. Arruda等人首先通过CycleGAN从源样本中生成与目标域相似的伪样本。^[1]

两个生成器：GN GD

两个判别器：DN DD

生成器 GN 将图像从白天映射到黑夜，而 GD 则是将图像从黑夜映射到白天。

判别器 DN 和 DD 分别判断图像是否属于该域。

通过损失 ℓ_1 来保证**循环一致性**，进而保证重建能力。

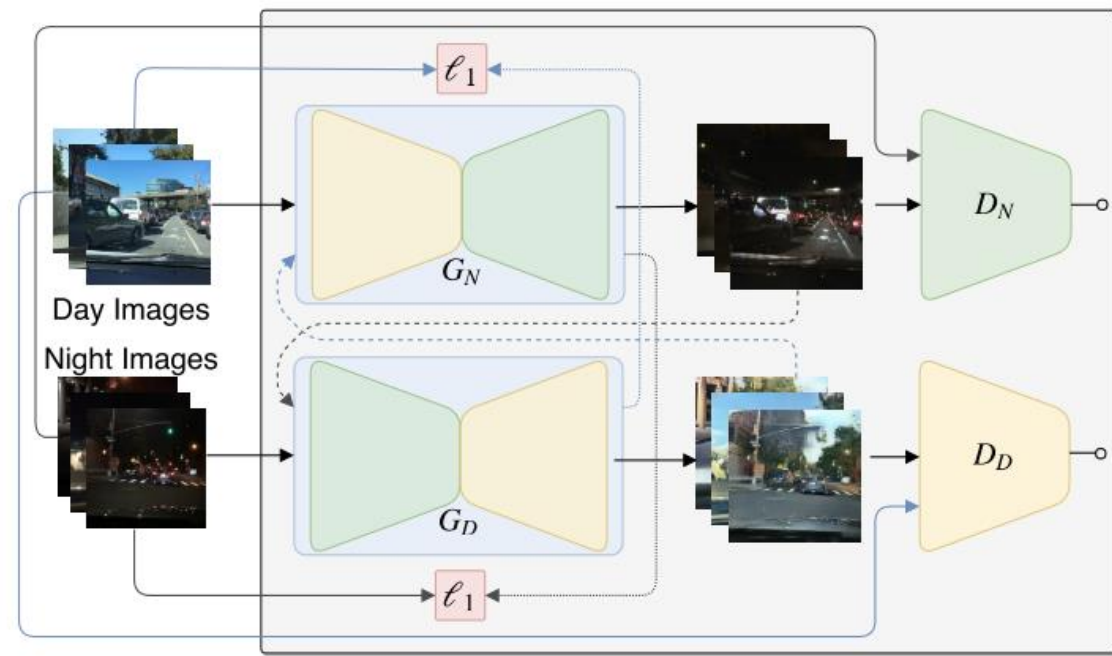


图5 CycleGAN架构在Vinicius等人文中的引用^[1]

[1] Arruda V F, Paixao T M, Berriel R F, et al. Cross-domain car detection using unsupervised image-to-image translation: From day to night[C]//2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019: 1-8.

1.2 相关工作

领域泛化相关研究

学习领域不变性方法和领域增强方法

领域不变性学习策略的目的是学习一种能够对齐多个源的特征空间的模型。

- Li等人提出了一种模型无关的训练算法，旨在提升模型应对领域偏移（domain shift）的能力。这种方法并不设计特定的模型，而是通过训练过程中的元优化目标（meta-optimization objective）来提高任意模型的泛化能力。

它的性能仍然取决于源域的多样性。如果源域数据不足以涵盖目标域的各种变化，模型的泛化能力仍会受到限制

Algorithm 1 Meta-Learning Domain Generalization

```
1: procedure MLDG
2:   Input: Domains  $\mathcal{S}$ 
3:   Init: Model parameters  $\Theta$ . Hyperparameters  $\alpha, \beta, \gamma$ .
4:   for ite in iterations do
5:     Split:  $\bar{\mathcal{S}}$  and  $\check{\mathcal{S}} \leftarrow \mathcal{S}$ 
6:     Meta-train: Gradients  $\nabla_{\Theta} = \mathcal{F}'_{\Theta}(\bar{\mathcal{S}}; \Theta)$ 
7:     Updated parameters  $\Theta' = \Theta - \alpha \nabla_{\Theta}$ 
8:     Meta-test: Loss is  $\mathcal{G}(\check{\mathcal{S}}; \Theta')$ .
9:     Meta-optimization: Update  $\Theta$ 

$$\Theta = \Theta - \gamma \frac{\partial(\mathcal{F}(\bar{\mathcal{S}}; \Theta) + \beta \mathcal{G}(\check{\mathcal{S}}; \Theta - \alpha \nabla_{\Theta}))}{\partial \Theta}$$

10:   end for
11: end procedure
```

图6 元学习领域泛化算法^[1]

元优化结合了元训练损失和元测试损失的梯度优化模型参数，使模型在两个阶段的梯度方向尽可能一致，提升模型泛化能力。

[1] Li D, Yang Y, Song Y Z, et al. Learning to generalize: Meta-learning for domain generalization[C]//Proceedings of the AAAI conference on artificial intelligence. 2018, 32(1).

1.2 相关工作

领域泛化相关研究

对于领域增强方法

- Xu等人提出了一种基于傅里叶变换的框架 (Fourier Augmented Co-Teacher, FACT) 来解决深度学习模型的领域泛化问题。方法的核心假设是傅里叶变换的相位信息包含了高阶语义，且不易受领域变化影响。^[2]

现有领域增强方法存在的问题

这些增强方法需要参考域，这限制了源域的多样性。

- Huang等人提出了一种基于频率空间域随机化 (Frequency Space Domain Randomization, FSDR) 的方法，用于领域泛化问题。使用网络分离域不变量和域特定频率分量，然后将域特定频率分量转移到其他辅助域。^[1]

FSDR通过使用离散余弦变换 (Discrete Cosine Transform, DCT) 将图像分解为不同的频率成分。将图像转换为频率空间，识别和随机化域变频率成分 (Domain Variant Frequency Components, DVFs)，同时保持域不变频率成分 (Domain Invariant Frequency Components, DIFs) 不变。

[1] Huang J, Guan D, Xiao A, et al. Fsd: Frequency space domain randomization for domain generalization[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 6891-6902.

[2] Xu Q, Zhang R, Zhang Y, et al. A fourier-based framework for domain generalization[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 14383-14392.

1.3 本文的创新

问题分析

领域适应 DA:

- 需要目标域

领域泛化 DG:

- 需要多样化的源域
- 存在域偏差问题
- 领域增强方法需要参考域

本文提出的创新

- **提出RST方法:** 在图像级将源图像随机变换为各种各样的新图像
- **提出基于领域的梯度加权模块GW:** 缓解领域偏差
- **将我们的RST和GW嵌入到一个通用的元学习框架中:** 提出解决DGOD问题的MLRT(Meta Learning on Randomized Transformations)模型

2

研究内容与方法

2.1 MLRT框架

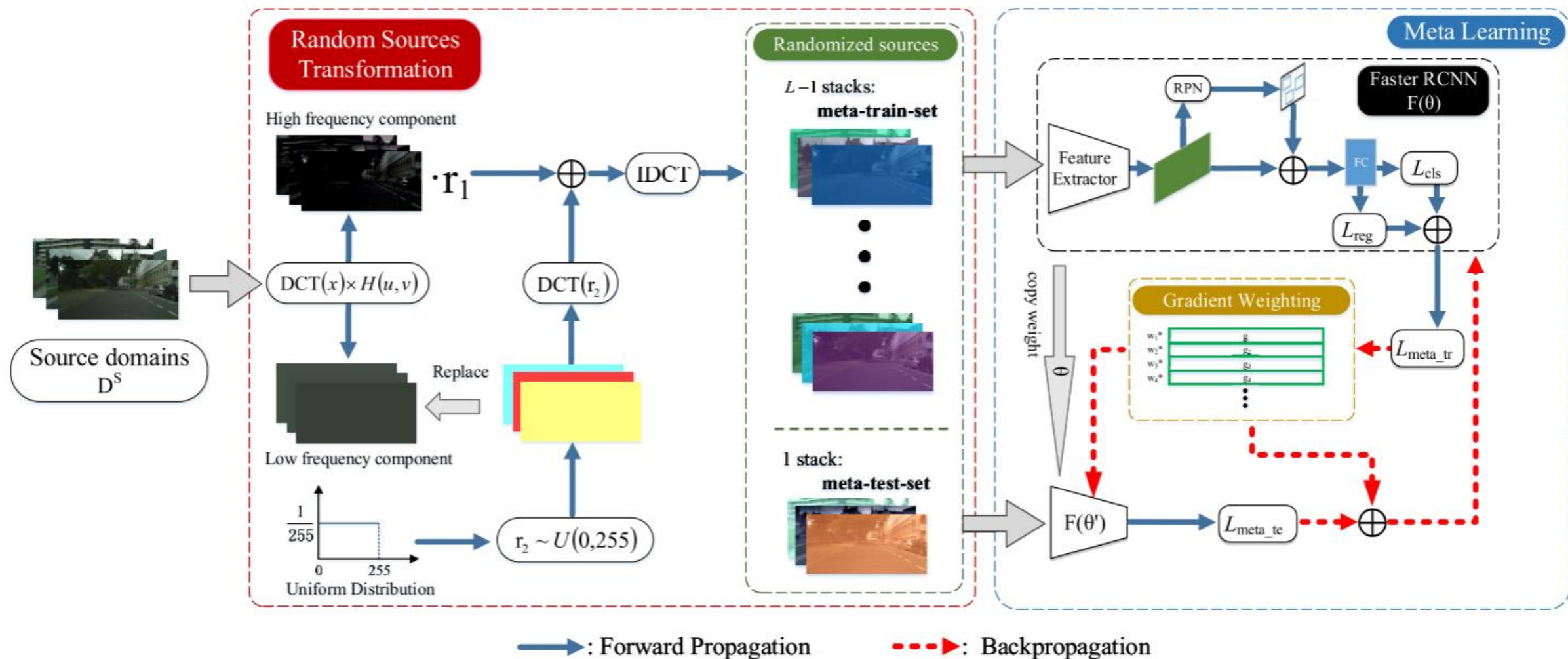


图7 MLRT框架

2.2 RST(Random Spectrum Transformation)

随机频谱变换

图像的**低频分量**包含许多**特定于域**的信息(例如背景颜色)

图像的**高频分量**包含许多**域不变**信息(即对象语义)

基于这以理念，我们使用高通滤波器 $H(u, v)$ 来分离图片 x 的语义信息和特定于领域的信息。

同时，为了打破对参考域的依赖。本文在不使用任何额外数据的情况下对低频信息进行随机替换。

$$\hat{x} = IDCT(r_1 DCT(x) \times H(u, v) + m(u, v)) \quad (1)$$

令(1)式 $\hat{x} = RLT(x)$ (Randomized Low-frequency-component Transformation)

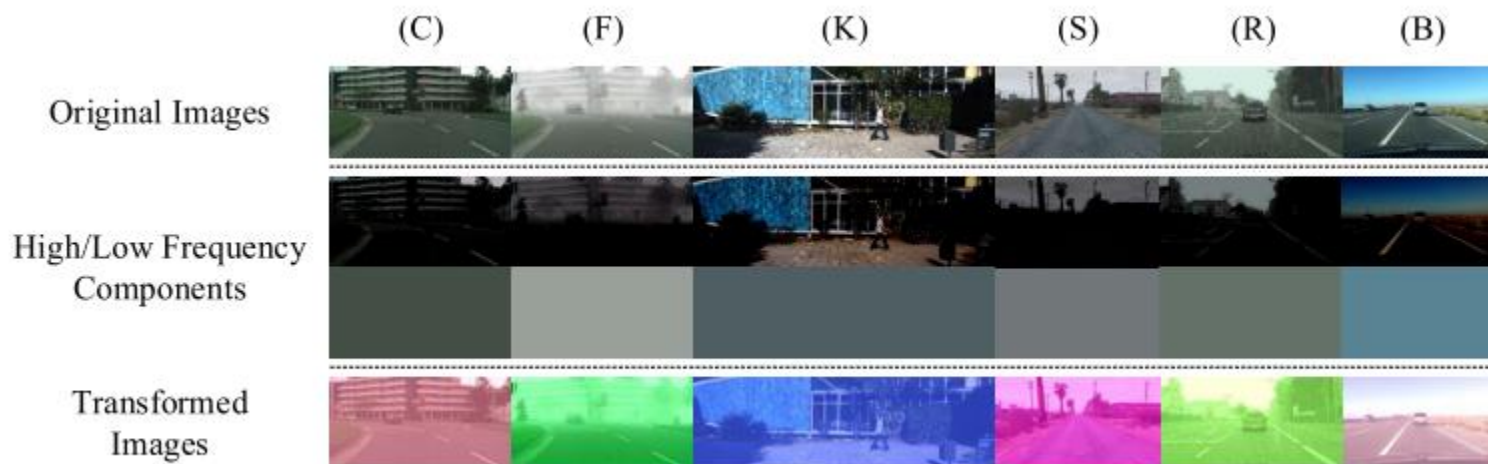


图8 图像低频分量和高频分量的可视化。行标表示城市景观(C)、雾城市景观(F)、KITTI (K)、雨城市景观(R)、SIM 10k (S)和BDD100k (B)

2.2 RST(Random Spectrum Transformation)

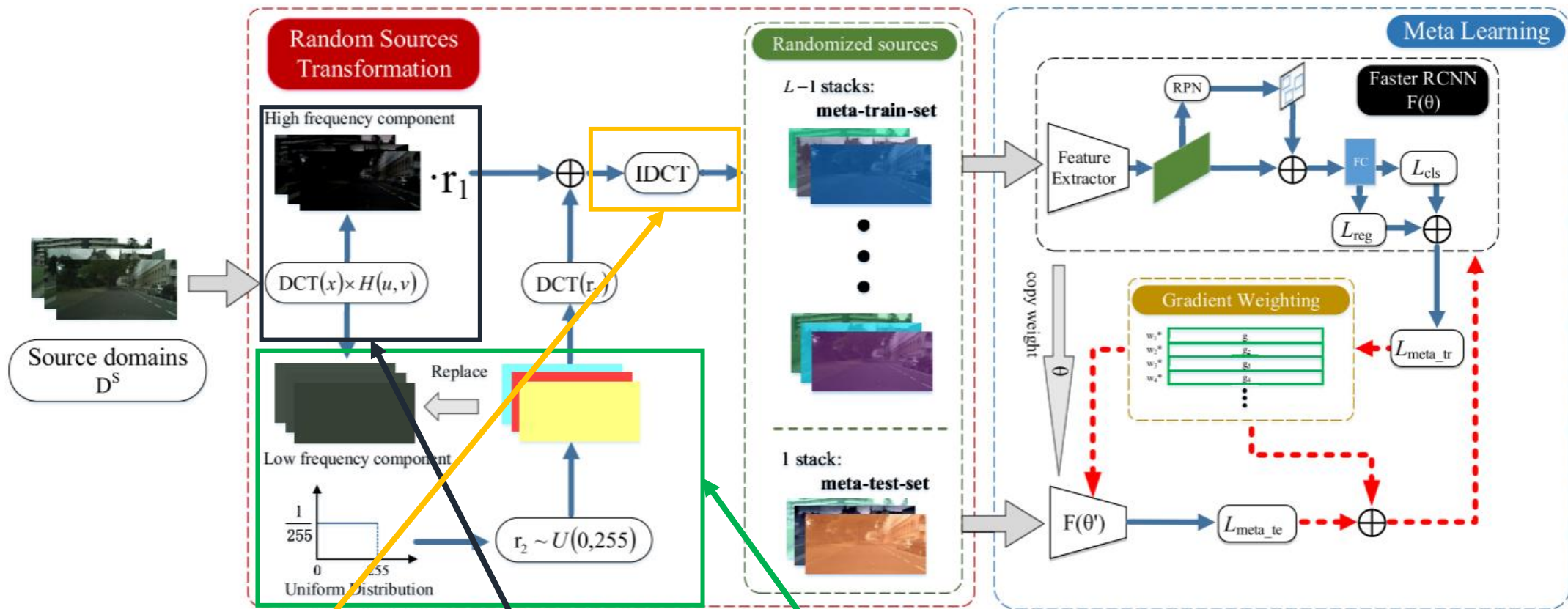


图9 MLRT框架

$$\hat{x} = IDCT(r_1 DCT(x) \times H(u, v) + m(u, v)) \quad (1)$$

$$r_1 \sim U(0.2, 1) \quad m(u, v) = DCT(r_2), r_2 \sim U(0, 255)$$

2.2 RST(Random Spectrum Transformation)

$$\hat{x} = IDCT(r_1 DCT(x) \times H(u, v) + m(u, v)) \quad (1)$$

将频谱逆变回图片

随机退化系数

图片频谱

提取图片高频信息

加入随机低频信息

$H(u, v)$ 表示高通滤波器定义式为:

$$H(u, v) = 1 - e^{-\frac{u^2 + v^2}{2R^2}} \quad (2)$$

其中R表示截止频率, 这里设 $R=1$

$$\begin{aligned} & \{D^1, \dots, D^N\} \\ D^j &= (x_i^j, y_i^j)_{i=1}^M \\ S^l &= (RLT(x_i), y_i)_{i=1}^M, 1 \leq l \leq L \end{aligned}$$

我们有L个由原始数据集D生成的增强数据集S, 每个S包含M个数据标签对, 故我们通过RLT变换成立了 $M \cdot L$ 个不同的随机域

2.3 GW(Gradient Weighting)

梯度加权

从梯度平衡的角度缓解域偏差问题

$\cos\theta_i$ 表示 (g_c, g_i) 的余弦相似度

$\cos\bar{\theta}_i$ 表示 (g_c, \bar{g}_i) 的余弦相似度

如果二者差小于阈值, 说明 D^i 域没有偏离其他域, $w_i = 1$

如果二者差大于阈值, 说明 D^i 域偏离

了其他域, $w_i = \frac{\|\bar{g}_i\|_2}{\|g_i\|_2}$

$$w_i = \begin{cases} 1, & \cos\bar{\theta}_i - \cos\theta_i \leq \delta \\ \frac{\|\bar{g}_i\|_2}{\|g_i\|_2}, & \cos\bar{\theta}_i - \cos\theta_i > \delta \end{cases}$$

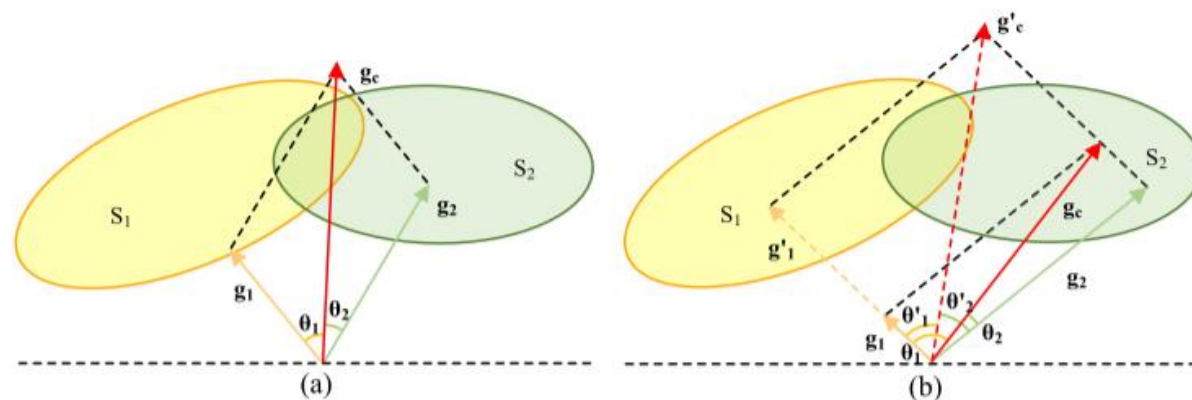


图10 训练阶段的两种梯度情况(a)表示无梯度偏差的情况(b)表示梯度和由 g_2 占据主导的情况

$$(3) \quad g' = \sum_{i=1}^L w_i g_i \quad (4)$$

2.3 GW(Gradient Weighting)

梯度加权

如图1(a)所示例子
这表示 g_c 不受 g_1 支配, 也不受 g_2 支配
即表示此时 $\cos\theta_1$ 和 $\cos\theta_2$ 很接近

当 $|\cos\theta_2 - \cos\theta_1| \leq \delta$ 时, 我们有

$$\begin{cases} w_1 = 1, & |\cos\theta_2 - \cos\theta_1| \leq \delta \\ w_2 = 1, & |\cos\theta_2 - \cos\theta_1| \leq \delta \end{cases} \quad (5)$$

而对于如图1(b)所示例子
此时 g_c 即受 g_2 支配

当 $\cos\theta_2 - \cos\theta_1 > \delta$ 时, 我们希望给 g_1 乘以一个较大的梯度 w_1 , 给 g_2 乘以一个较小的梯度 w_2

以此我们得到 $g'_c = g'_1 + g'_2$, $g'_1 = w_1 g_1$ $g'_2 = w_2 g_2$

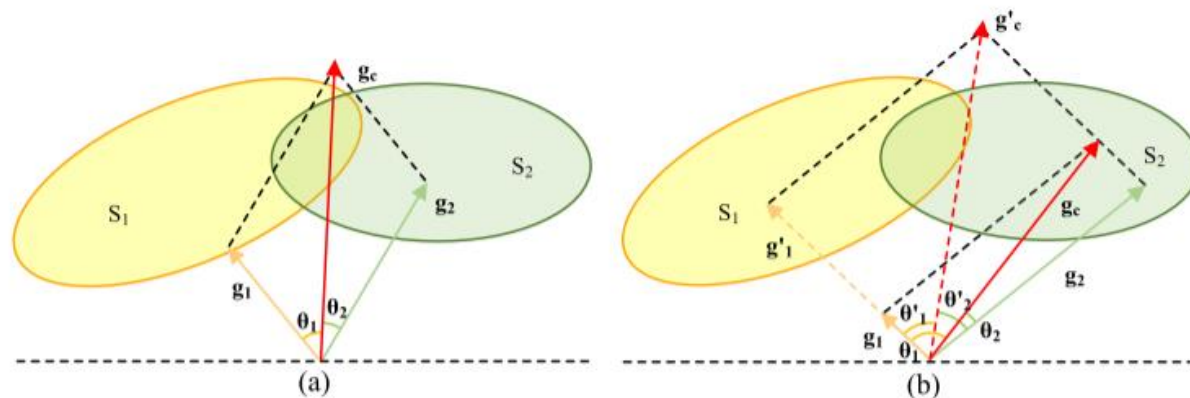


图10 训练阶段的两种梯度情况(a)表示无梯度偏差的情况(b)表示梯度和由 g_2 占据主导的情况

2.3 GW(Gradient Weighting)

梯度加权

目的：希望S1和S2平衡

$$\cos\theta'_1 = \cos\theta'_2$$

而余弦相似度的计算公式为

$$\cos\theta'_i = \frac{g'_c \cdot g'_i}{\|g'_c\|_2 \times \|g'_i\|_2}$$

$$\frac{g'_c \cdot g'_1}{\|g'_c\|_2 \times \|g'_1\|_2} = \frac{g'_c \cdot g'_2}{\|g'_c\|_2 \times \|g'_2\|_2}$$

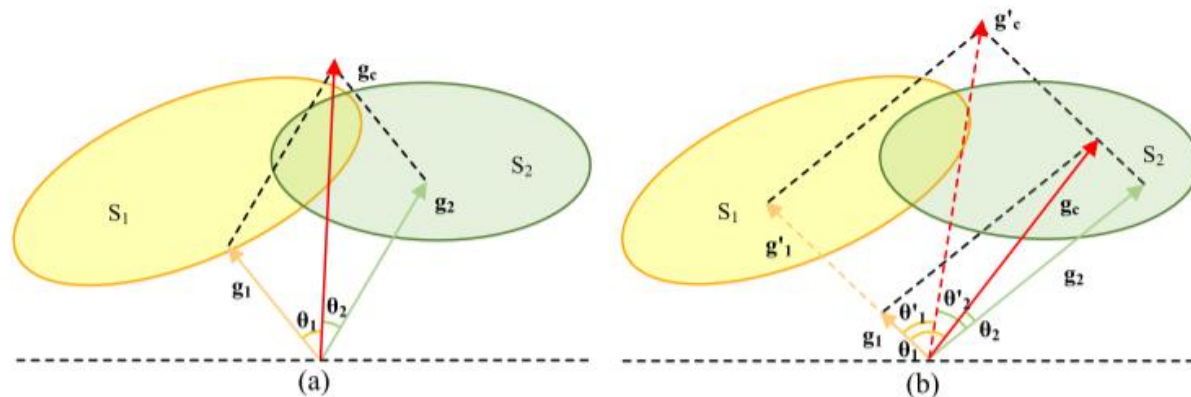
再将 $g'_c = g'_1 + g'_2$, $g'_1 = w_1 g_1$ $g'_2 = w_2 g_2$ 带入，我们便可得到(8)式

$$w_1 \|g_1\|_2 = w_2 \|g_2\|_2$$

再取 $w_2 = 1$ 我们最后便能得出(9)式

$$(8) \quad \begin{cases} w_1 = \frac{\|g_2\|_2}{\|g_1\|_2}, & \cos\theta_2 - \cos\theta_1 > \delta \\ w_2 = 1, & \cos\theta_2 - \cos\theta_1 > \delta \end{cases} \quad (9)$$

(6)



(7)

图10 训练阶段的两种梯度情况(a)表示无梯度偏差的情况(b)表示梯度和由g2占据主导的情况

2.3 GW(Gradient Weighting)

梯度加权

$$\begin{cases} w_1 = \frac{\|g_2\|_2}{\|g_1\|_2}, & \cos\theta_2 - \cos\theta_1 > \delta \\ w_2 = 1, & \cos\theta_2 - \cos\theta_1 \leq \delta \end{cases} \quad (9)$$

当 $\cos\theta_1 - \cos\theta_2 > \delta$ 我们同理可得(10)

$$\begin{cases} w_1 = 1, & \cos\theta_1 - \cos\theta_2 > \delta \\ w_2 = \frac{\|g_1\|_2}{\|g_2\|_2}, & \cos\theta_1 - \cos\theta_2 \leq \delta \end{cases} \quad (10)$$

由此, 对于任意的 g_i 我们的(3)式子成立

结合(5)(9)(10)我们可以得出:
对于 g_1 的权重可以表示为

$$w_1 = \begin{cases} 1, & \cos\theta_2 - \cos\theta_1 \leq \delta \\ \frac{\|g_2\|_2}{\|g_1\|_2}, & \cos\theta_2 - \cos\theta_1 > \delta \end{cases} \quad (11)$$

同理, 对于 g_2 的权重可以表示为

$$w_2 = \begin{cases} 1, & \cos\theta_1 - \cos\theta_2 \leq \delta \\ \frac{\|g_1\|_2}{\|g_2\|_2}, & \cos\theta_1 - \cos\theta_2 > \delta \end{cases} \quad (12)$$

$$w_i = \begin{cases} 1, & \cos\bar{\theta}_i - \cos\theta_i \leq \delta \\ \frac{\|\bar{g}_i\|_2}{\|g_i\|_2}, & \cos\bar{\theta}_i - \cos\theta_i > \delta \end{cases} \quad (3)$$

2.4 Meta Learning on Randomized Transformations

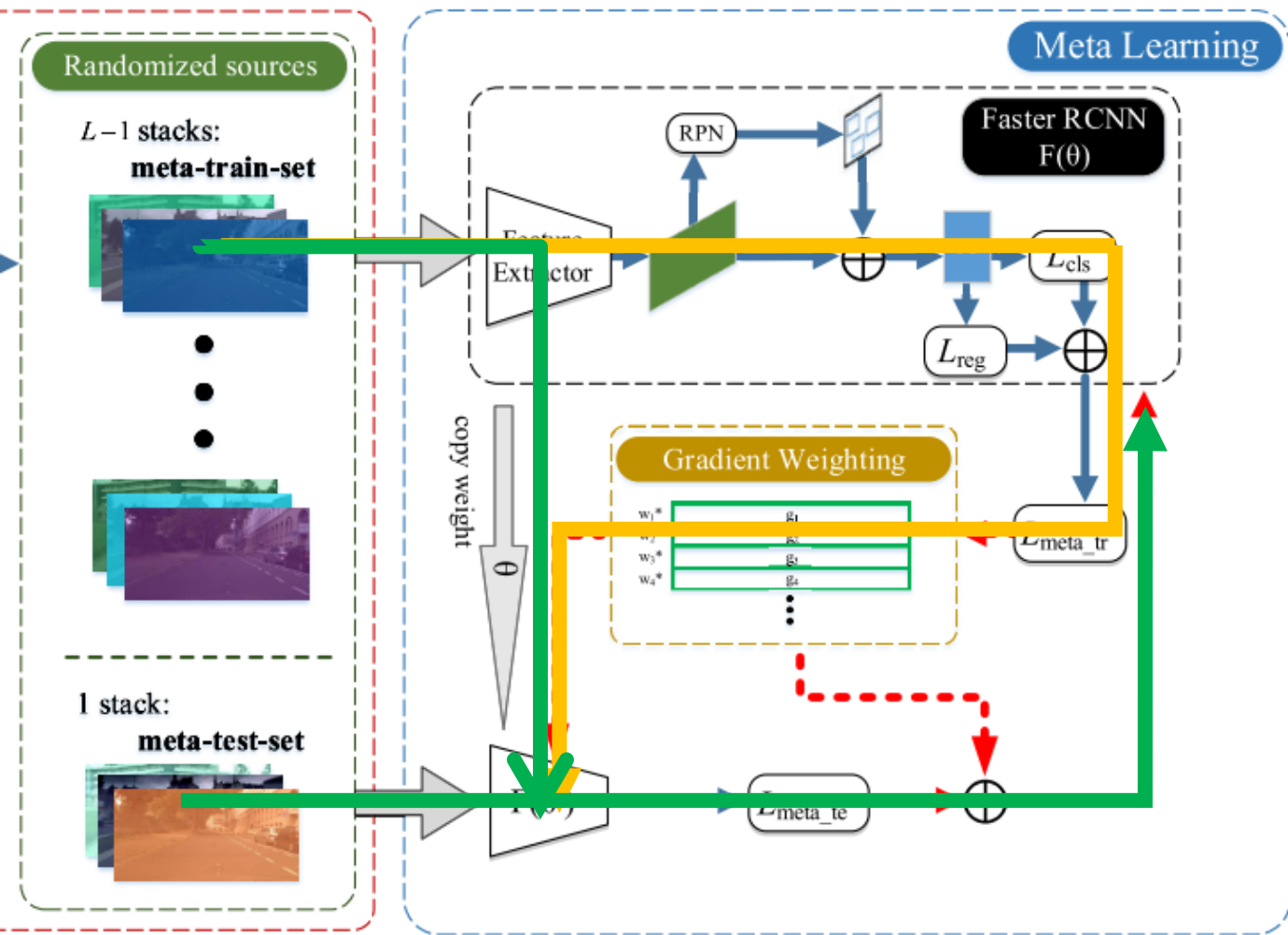


图11 梯度加权GW与用于随机源上的目标检测的元学习范例

基于随机变换的元学习

- 我们在元训练集上训练我们的模型，并将我们模型的参数 θ 更新为 θ'
- $$\theta' = \theta - \alpha \cdot \sum_{l=1}^{L-1} w_l \frac{\partial L_{detect}(D_l^{train}, \theta)}{\partial \theta} \quad (13)$$
- 我们对更新后的模型 θ' 在meta-test-set上测试，同时使用meta-test-set和meta-train-set上的损失对原始模型 θ 进行更新

$$\theta' = \theta - \alpha \cdot \sum_{l=1}^{L-1} w_l \frac{\partial L_{detect}(D_l^{train}, \theta)}{\partial \theta} \quad (13)$$

$$\begin{aligned} \theta = \theta - lr \cdot \sum_{l=1}^{L-1} w_l \frac{\partial L_{detect}(D_l^{train}, \theta)}{\partial \theta} \\ - lr \frac{\partial L_{detect}(D^{test}, \theta')}{\partial \theta} \end{aligned} \quad (14)$$

2.5 MLRT减少域偏差的分析

使用元学习来结合RST和GW模块

- [1]提出了一个用于域泛化的元学习范式，显示出了强大的潜力。
- 元学习与RST和GW相辅相成。
- 借助元学习框架的理论支持。

遵循其主要的设置并进行改进

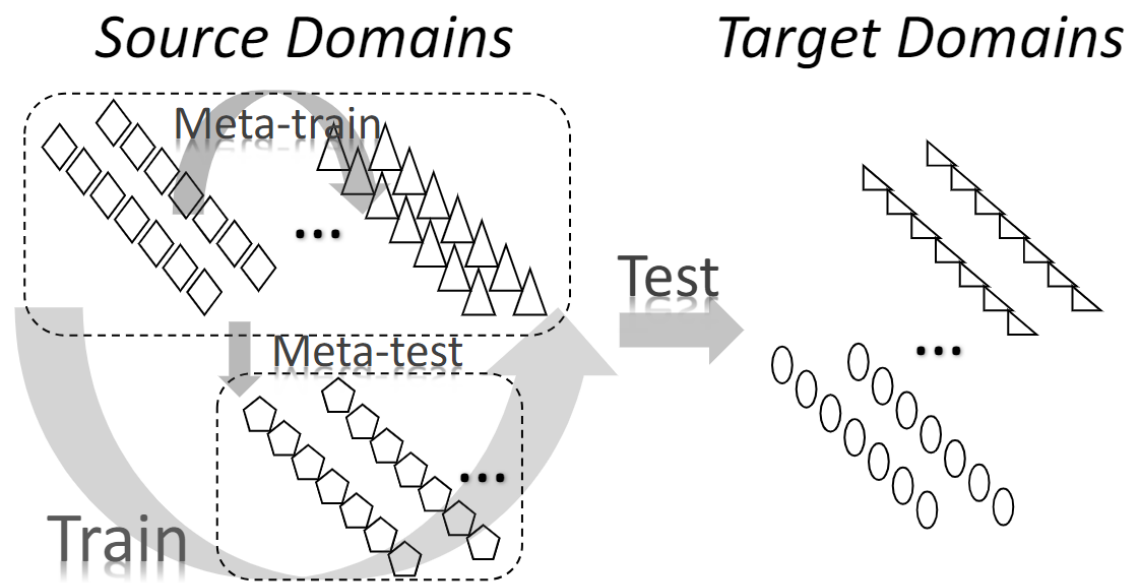


图12 元学习领域泛化方法的图示。符号代表不同的数据域^[1]

[1] Learning to Generalize: Meta-Learning for Domain Generalization, AAAI 2023.

2.5 MLRT减少域偏差的分析

损失函数分析

$$L_{meta} = L_{meta_tr}(D^{train}, \theta) + L_{meta_te}(D^{test}, \theta') \quad (15)$$



$$L_{meta} = L_{meta_tr}(D^{train}, \theta) + L_{meta_te}(D^{test}, \theta) - \alpha \cdot g_{train} \cdot g_{test} \quad (17)$$

最大化meta-train损失和meta-test损失的梯度内积:

$$g_{train} \cdot g_{test} = \sum_{l=1}^{L-1} g_l \cdot g_{test} \quad \longrightarrow \quad g_{train} \cdot g_{test} = \sum_{l=1}^{L-1} \underline{w_l g_l^{rand}} \cdot g_{test}$$

存在的问题: **域偏差**

如果其中一个 g_l 比其他梯度大得多或者几个梯度方向太接近, 则梯度的内积很容易被某些特定域主导, 这意味着优化方向由特定域主导

在MLRT中:

2.6 泛化的理论见解

泛化的理论见解

目标域上假设 h 的风险

- 真实标签函数 f : 对于包含数据和标签对 (x, y) 的域 D , f 是确定性函数, 表示对每个输入 x 的到真实标签的映射且有 $y = f(x)$ 。

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

- 假设函数 H : 存在 $h(x)$, 且 $h \in H$, 其中 H 是一组候选假设。

$$h : \mathcal{X} \rightarrow \mathcal{Y}$$

- 域 D 的假设 h 风险: $\epsilon(h) = \mathbb{E}_{x \in D}(L(h(x), f(x)))$ (18)

2.6 泛化的理论见解

泛化的理论见解

目标域上假设 h 的风险

- 在域泛化设置中，对于源域和目标域，同样存在这样的风险。对于若干个源域 $\{D_1, \dots, D_N\}$ 和一个目标域 D_T ，有源域风险 ϵ_S 和目标域风险 ϵ_T 。
- 域泛化中目标域的风险表示^{[1][2][3]}：

$$\epsilon_T(h) \leq \sum_{n=1}^N \alpha_n \epsilon_S^n(h) + \frac{\rho}{2} + \lambda_\alpha \quad (20)$$

- H散度：
$$d_{\mathcal{H}}[\mathcal{D}_S, \mathcal{D}_T] = 2 \sup_{\eta \in \mathcal{H}} |\Pr_{x \sim \mathcal{D}_S} [\eta(x) = 1] - \Pr_{x \sim \mathcal{D}_T} [\eta(x) = 1]|$$

[1] Generalizing to unseen domains via distribution matching. (arXiv 2019)

[2] Adversarial Multiple Source Domain Adaptation. (NIPS 2018)

[3] Algorithms and Theory for Multiple-Source Adaptation. (NIPS 2018)

2.6 泛化的理论见解

泛化的理论见解

目标域上假设 h 的风险

最优假设 h^* 的风险：理想假设的联合误差

$$\epsilon_T(h) \leq \epsilon_S(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) + \underbrace{(\epsilon_S(h^*) + \epsilon_T(h^*))}_{[1]}$$



$$\epsilon_T(h) \leq \sum_{n=1}^N \alpha_n \epsilon_S^n(h) + \frac{\rho}{2} + \lambda_\alpha \quad (20)$$

$$\rho = \sup_{D^i, D^j \in \{D^n\}_{n=1}^N} d_{\mathcal{H}}(D^i, D^j)$$

$$\mathcal{H}\Delta\mathcal{H} = \{h(x) \oplus h'(x) | h, h' \in \mathcal{H}\}$$

引理^[1]:

$$|\epsilon_S(h, h') - \epsilon_T(h, h')| \leq \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T)$$

$$D^T = \sum_{n=1}^N \alpha_n D^n \quad (19)$$

$$\operatorname{argmin}_{\pi_1, \dots, \pi_{N_S}} d_{\mathcal{H}} \left[\mathcal{D}_U, \sum_{i=1}^{N_S} \pi_i \mathcal{D}_S^i \right] \quad [2]$$

$$d_{\mathcal{H}}[\mathcal{D}', \mathcal{D}''] \leq \epsilon \quad [2]$$

[1]. A theory of learning from different domains (Mach Learn 2010)

[2] Generalizing to unseen domains via distribution matching. (arXiv 2019)

[3] Adversarial Multiple Source Domain Adaptation. (NIPS 2018)

[4] Algorithms and Theory for Multiple-Source Adaptation. (NIPS 2018)

2.6 泛化的理论见解

泛化的理论见解

目标域上假设 h 的风险

$$\epsilon_T(h) \leq \sum_{n=1}^N \alpha_n \epsilon_S^n(h) + \frac{\rho}{2} + \lambda_\alpha \quad (20)$$

$\left[\begin{array}{c} \rho = \sup_{D^i, D^j \in \{D^n\}_{n=1}^N} d_{\mathcal{H}}(D^i, D^j) \end{array} \right]$

$$L_{meta} = L_{meta_tr}(D^{train}, \theta) + L_{meta_te}(D^{test}, \theta) - \alpha \cdot g_{train} \cdot g_{test} \quad (17)$$

$$a \cdot b = ||a||_2 ||b||_2 \cos(\delta)$$

- 上界的第一项：对应loss的第一项和第二项
- 上界的第二项：对应loss的第三项
- 上界的第三项：忽略不计

3

实验结果与分析

3.1 实验结果与分析

数据集

- **Cityscapes数据集**: 主要包含白天街道上的场景, 2,975张图片用于训练, 500张图片用于测试;
- **Foggy Cityscapes数据集**: 包含通过在Cityscapes中添加浓雾而创建的恶劣天气图像, 2,975张图片用于训练, 500张图片用于测试。
- **Rain Cityscape**: Rain Cityscape是通过在Cityscape中添加雨条纹和雾得到的, 包含 9,432 个训练图像和 1,188 个测试图像。
- **KITTI**: 是一个自动驾驶数据集, 其中包含7,481张图像用于训练, 7518张图像用于测试。该数据集的主要场景是中型城市的街道, 并且该数据集中的目标尺寸非常小。



图14 Cityscapes数据集 (上) 和Foggy Cityscapes数据集 (下)



图15 Rain Cityscapes数据集



图16 KITTI数据集

3.1 实验结果与分析

数据集

- **SIM10k**: 包含计算机3D模型的渲染图像，包括10,000张有标注图像，其中5,000张用作训练集，其余用作测试集。由于SIM10k中每个类别的实例数量极不平衡，在本文实验中仅考虑类别{car}
- **BDD100k**: 用于自动驾驶的大规模数据集。在本文实验中使用总共 100,000 张图像中的 10,000 张作为训练集，10,000 张图像作为测试集。

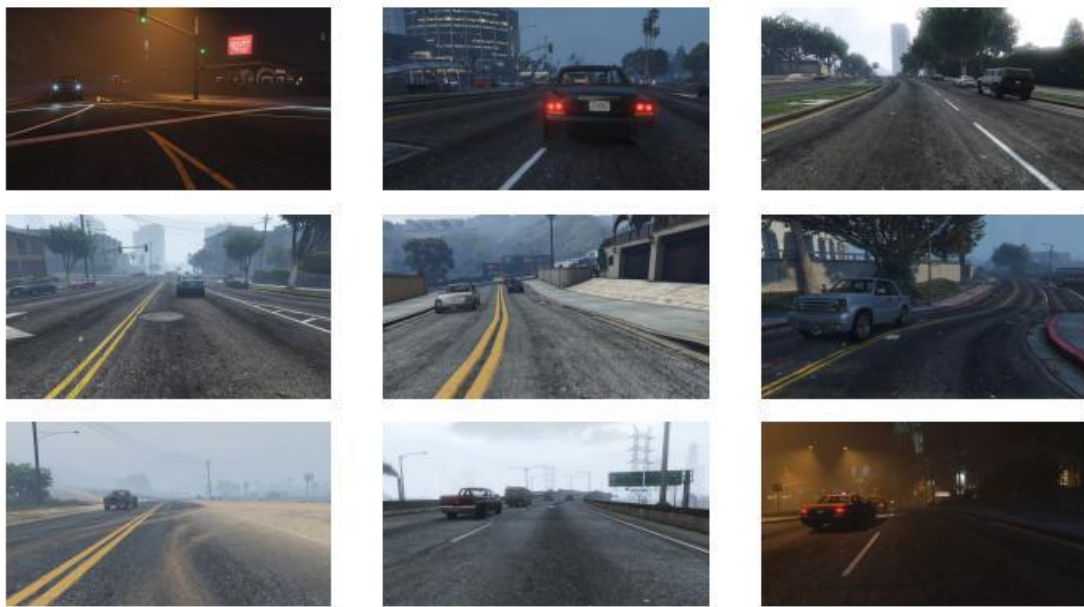


图17 SIM10k数据集



图18 BDD100k数据集

3.1 实验结果与分析

实施细节

- **数据集表示**: SIM 10k、Cityscapes、Foggy Cityscapes、BDD100k、KITTI、Rain Cityscape分别缩写为S、C、F、B、K、R。
- **域泛化设置**: 单源域泛化;
- **图像预处理**: 调整短边尺寸为600像素, 保持长宽比不变
- **baseline**: 使用元学习框架的Faster R-CNN^[1]
- **backbone**: 在ImageNet上预训练的vgg16^[2]
- **learning rate**: 0.002 \rightarrow 0.0002
- **optimization**: SGD
- **epoch**: 10
- **元学习的设置**: 在每个batch中, 使用 L- 1张图像进行元训练, 使用1张图像进行元测试。

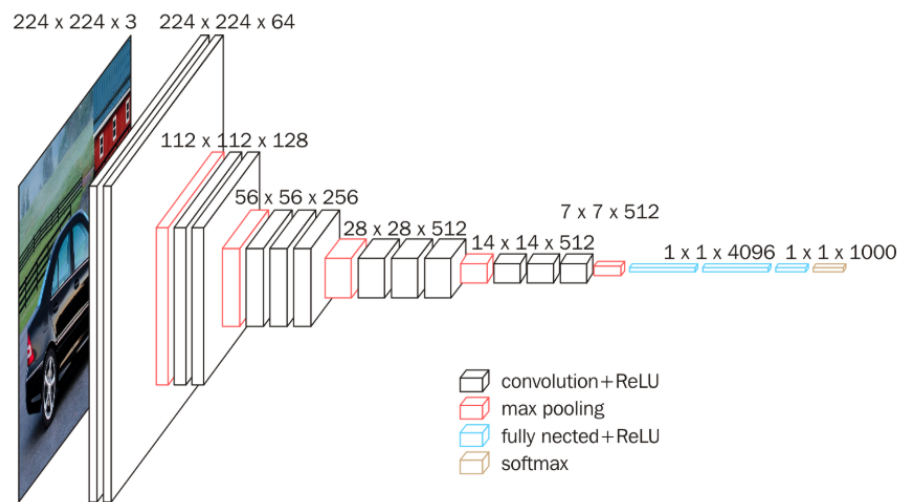


图19 VGG16结构示意图

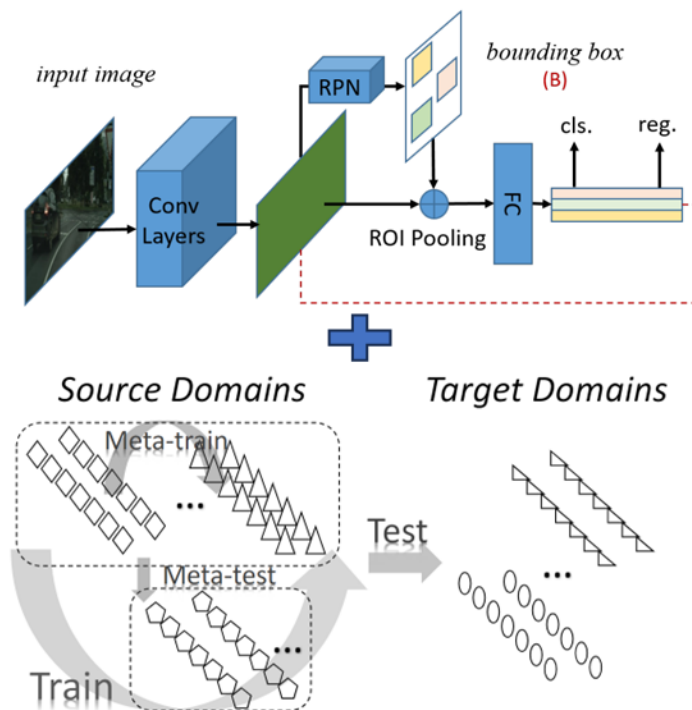


图20 baseline: 使用元学习框架的Faster R-CNN

[1] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. (TPAM 2017)

[2] Very deep convolutional neural network based image classification using small training sample size. (ACPR 2015)

3.1 实验结果与分析

实施细节

- 评价指标:

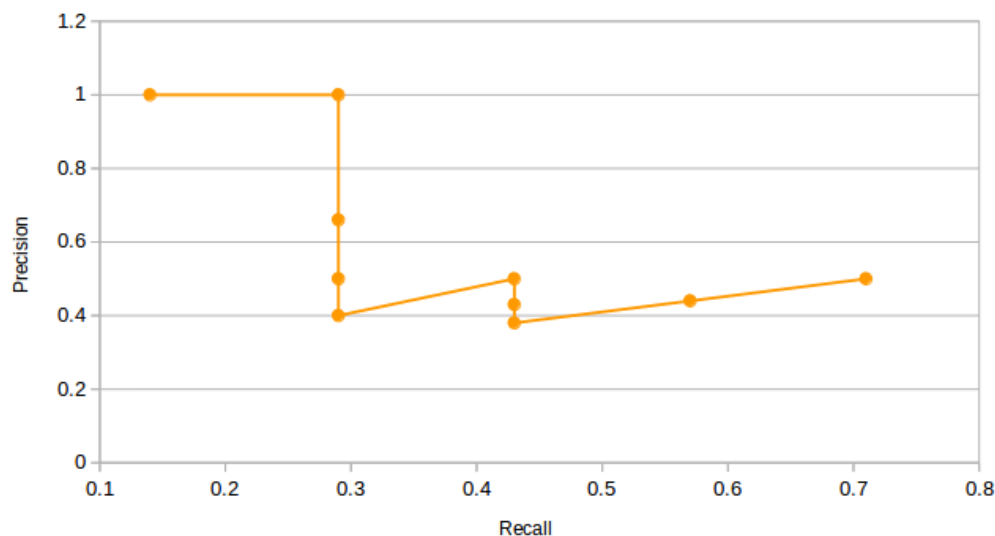
- (1) 平均精度(mAP) , 是对所有类别的AP (average precision) 取平均值

- (2) 交并比 (Intersection over Union, IoU)

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

P-R curve



$$IOU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{area of overlap}}{\text{area of union}}$$

求面积

求平均

AP

mAP

3.1 实验结果与分析

与各种DG方法进行比较

表1 在 C、F、B 上训练的五种 DGOD 方法的结果 (%). 注意, * 表示该方法是使用已发布的代码重现的。

Source	Target	Method	person	rider	car	truck	bus	train	motor	bike	mAP
Cityscape	Fogy	Source-only	26.4	40.1	35.9	14.5	27.7	9.2	25.0	33.5	26.5
		DSU*	26.7	41.2	40.9	13.0	26.5	9.1	24.0	33.3	26.8
		Mixstyle*	27.2	41.6	41.4	16.0	30.3	12.5	22.1	32.5	28.0
		MLDG*	29.7	42.5	43.1	22.0	29.8	2.4	27.3	31.7	28.6
		FACT*	26.2	41.2	35.9	13.6	27.7	3.0	23.3	31.3	25.3
		FSDR*	31.2	44.4	43.3	19.3	36.6	11.9	27.1	34.1	31.0
		NP*	31.9	45.9	46.5	24.3	37.7	12.6	25.1	36.6	32.6
		MLRT(Ours)	34.8	48.1	49.1	24.3	43.6	21.9	30.7	39.0	36.4
	Rain	Source-only	26.9	62.2	49.7	10.4	72.9	-	19.8	36.3	39.7
		DSU*	25.2	62.6	50.7	9.5	59.6	-	18.9	32.6	37.0
		Mixstyle*	26.0	62.2	51.2	6.3	69.2	-	19.2	33.0	38.2
		MLDG*	26.2	61.2	52.7	17.9	79.9	-	31.4	25.7	42.1
		FACT*	25.9	62.4	48.8	17.4	62.0	-	29.1	33.5	39.9
		FSDR*	27.7	62.7	52.7	14.8	76.2	-	28.4	37.1	42.8
		NP*	26.4	62.7	53.0	11.7	71.7	-	25.7	35.0	40.9
		MLRT(Ours)	28.9	61.7	57.5	18.7	78.4	-	26.1	35.3	43.8
	BDD100K	Source-only	30.4	29.6	41.3	16.4	20.5	-	11.8	27.2	25.3
		DSU*	28.2	30.3	43.4	18.2	20.3	-	13.1	28.4	26.0
		Mixstyle*	26.2	28.2	40.3	17.4	19.6	-	11.8	25.8	24.2
		MLDG*	24.6	23.3	39.0	15.4	12.7	-	11.4	20.9	21.0
		FACT*	31.1	29.3	42.0	20.5	19.8	-	11.2	28.1	26.0
		FSDR*	30.7	29.0	45.4	18.5	20.0	-	12.1	27.8	26.2
		NP*	31.7	33.0	45.3	20.4	20.3	-	12.9	30.2	27.7
		MLRT(Ours)	32.8	29.2	45.0	19.4	19.5	-	12.2	27.4	26.5
BDD100K	Cityscape	Source-only	31.6	33.2	53.2	34.3	37.6	-	20.1	25.0	33.6
		DSU*	31.5	32.1	53.1	33.7	40.2	-	19.0	25.0	33.5
		Mixstyle*	31.0	33.2	53.0	35.4	39.7	-	18.4	25.9	33.8
		MLDG*	16.1	21.0	52.4	27.2	34.8	-	7.2	13.9	24.7
		FACT*	30.9	32.3	53.2	33.3	33.2	-	17.9	26.0	32.4
		FSDR*	31.2	32.6	53.1	32.8	33.7	-	17.1	26.0	32.4
		NP*	31.2	33.5	53.1	37.2	33.2	-	16.9	26.4	33.1
		MLRT(Ours)	34.6	33.3	53.2	36.9	35.7	-	18.5	26.8	34.1
	Fogy	Source-only	26.5	30.4	46.2	20.6	28.6	-	15.0	23.1	27.2
		DSU*	26.0	30.1	45.9	21.1	33.1	-	17.1	21.0	27.8
		Mixstyle*	26.5	31.9	43.8	22.8	33.1	-	18.7	23.8	28.7
		MLDG*	9.1	12.8	37.3	19.1	18.4	-	9.5	13.3	17.1
		FACT*	22.7	27.9	40.7	18.6	22.9	-	14.9	22.6	24.3
		FSDR*	28.0	32.3	44.1	24.4	26.5	-	14.9	24.5	27.8
		NP*	28.6	33.9	48.8	25.3	31.1	-	15.0	25.1	29.7
		MLRT(Ours)	29.8	33.9	50.1	25.7	33.4	-	18.8	26.6	31.2
	Rain	Source-only	22.1	43.2	58.0	20.6	72.4	-	3.3	20.6	34.3
		DSU*	21.7	37.5	56.4	18.1	70.4	-	11.0	17.2	33.2
		Mixstyle*	21.8	44.9	56.3	21.0	72.7	-	10.5	20.2	35.3
		MLDG*	9.1	16.7	41.2	15.8	52.3	-	0.0	5.0	20.0
		FACT*	18.0	40.4	52.0	23.2	72.2	-	10.5	20.9	33.9
		FSDR*	22.6	43.3	56.4	26.5	67.6	-	6.9	19.5	34.7
		NP*	23.3	44.4	58.2	29.4	66.9	-	3.7	17.5	34.8
		MLRT(Ours)	19.6	47.0	58.6	29.1	69.9	-	2.7	20.2	35.3

Fogy	Cityscape	Source-only	32.7	43.2	50.1	14.0	37.5	1.8	24.9	35.3	29.9
		DSU*	29.2	41.1	44.4	16.0	32.7	2.6	21.2	29.0	27.0
		Mixstyle*	29.9	41.9	44.5	17.4	35.2	2.4	23.8	32.2	28.4
		MLDG*	33.1	43.0	49.2	16.9	36.5	1.9	27.1	35.4	30.4
		FACT*	31.8	42.5	50.5	19.5	34.4	1.8	25.7	34.3	30.0
		FSDR*	32.2	41.9	50.5	18.8	39.8	4.2	28.4	34.6	31.3
		NP*	33.3	45.4	51.9	23.2	44.9	13.6	25.8	36.0	34.3
		MLRT(Ours)	31.9	43.1	51.2	19.6	38.4	3.6	31.2	37.0	32.0
	Rain	Source-only	27.0	61.3	53.1	21.4	67.1	-	5.7	33.4	38.4
		DSU*	24.5	58.8	52.2	11.9	61.9	-	11.8	26.9	35.4
		Mixstyle*	25.5	58.9	52.4	16.7	60.4	-	16.4	30.3	37.2
		MLDG*	27.0	56.5	52.8	19.2	62.5	-	19.6	32.6	38.6
		FACT*	25.7	60.6	52.5	14.1	71.2	-	14.1	32.7	38.7
		FSDR*	28.1	60.8	53.3	32.6	65.7	-	11.7	33.4	40.8
		NP*	26.1	58.1	52.6	8.5	66.9	-	16.8	30.0	37.0
		MLRT(Ours)	27.3	61.7	53.4	20.7	78.2	-	15.7	31.0	41.1
	BDD100K	Source-only	20.4	21.7	31.2	9.9	8.7	-	10.1	20.6	17.5
		DSU*	18.8	19.9	31.1	8.9	11.3	-	10.6	19.5	17.2
		Mixstyle*	20.7	22.6	32.0	9.8	13.0	-	10.5	20.8	18.5
		MLDG*	21.2	20.9	31.8	9.8	9.9	-	10.5	21.8	18.0
		FACT*	23.5	25.3	33.8	11.7	12.4	-	11.0	24.2	20.2
		FSDR*	23.9	22.7	36.9	11.9	13.9	-	10.2	23.6	20.4
		NP*	27.5	28.3	40.7	14.2	15.4	-	12.1	26.4	23.5
		MLRT(Ours)	24.9	24.0	37.7	12.7	11.8	-	11.3	24.4	21.0

- 总体情况: MLRT 总是比其他方法取得更好结果
- 特例: Cityscapes 和 BDD100k 作为目标域时
- 比baseline更差的情况: MLDG的C→B、FACT的C→F、FSDR的B→C、Mixstyle 为 C→R、DSU 为 C→R、NP 为 F→R

3.1 实验结果与分析

与其他方法进行比较

表2 从 C 到 F 的 DG 和 DA 实验结果 (%). DAOD方法在C上训练, 未标记F, 多域设置下的DGOD方法在C和B上训练, 单域设置下的DGOD方法在C上训练。

Methods		Dataset	person	rider	car	truck	bus	train	motor	bike	mAP
Source-only		Single S	26.4	40.1	35.9	14.5	27.7	9.2	25.0	33.5	26.5
DA	DAF[17]	Single Source & Target without labels	31.6	43.6	42.8	23.6	41.3	21.2	28.9	32.6	33.2
	SW-DA[21]		31.8	44.3	48.9	21.0	43.8	28.0	28.9	35.8	35.3
	SC-DA[24]		33.8	42.1	52.1	26.8	42.5	26.5	29.2	34.5	35.9
	MTOR[25]		30.6	41.4	44.0	21.9	38.6	40.6	28.3	35.6	35.1
	ICR-CCR[39]		32.9	43.8	49.2	27.2	45.1	36.4	30.3	34.6	37.4
	Coarse-to-Fine[54]		34.0	46.9	52.1	30.8	43.2	29.9	34.7	37.4	38.6
	GPA[22]		32.9	46.7	54.1	24.7	45.7	41.1	32.4	38.7	39.5
	Center-Aware[19]		41.5	43.6	57.1	29.4	44.9	39.7	29.0	36.1	40.2
DG	DIDN[50]	Multi S	31.8	38.4	49.3	27.7	35.7	26.5	24.8	33.1	33.4
	MLDG*[29]	Single S	32.2	41.7	38.9	19.2	33.0	9.1	23.5	36.3	29.2
	FACT*[32]		26.2	41.2	35.9	13.6	27.7	3.0	23.3	31.3	25.3
	FSDR*[31]		31.2	44.4	43.3	19.3	36.6	11.9	27.1	34.1	31.0
	MLRT(L=6)		34.1	49.7	47.9	25.9	45.1	29.1	32.0	39.4	37.9
Oracle		Target	37.8	47.4	53.0	31.6	52.9	34.3	37.0	40.6	41.8

3.1 实验结果与分析

DGOD 方法与 DAF 比较

表3 从领域 C 到 F、B、R、S 和 K 的六种方法在共享类别 {Car} 上的结果 (%). DGOD方法的训练过程中仅使用数据集C, 而DAF方法的训练过程中使用C和未标记的F。最好的结果以粗体突出显示, 比下限更差的结果以下划线突出显示

Method	F	R	B	S	K
Source-only	35.9	49.6	41.3	39.2	72.2
DAF	40.9	<u>48.3</u>	<u>37.3</u>	<u>33.0</u>	<u>66.1</u>
MLGD	43.0	<u>52.7</u>	<u>39.4</u>	<u>37.2</u>	<u>63.9</u>
FACT	<u>35.8</u>	<u>48.8</u>	42.0	41.2	73.2
FSDR	43.3	52.7	45.4	42.2	73.8
MLRT(Ours)	49.1	57.5	45.5	43.6	73.4

与DG 图像分类的结果比较

表4 DG 图像分类的结果, 在 PACS 数据集集中的三个域上进行训练并在其余域上进行测试

Method	Art	Cartoon	Photo	Sketch	Avg
Mixstyle	84.1	78.8	96.1	75.9	83.7
DSU	83.6	79.6	95.8	77.6	84.1
FACT	85.4	78.4	95.2	79.2	84.5
MLRT(Ours)	84.6	80.4	95.2	83.2	85.9

3.1 实验结果与分析

消融实验

- 实验设置：在C上训练MLRT并在F、R、B测试

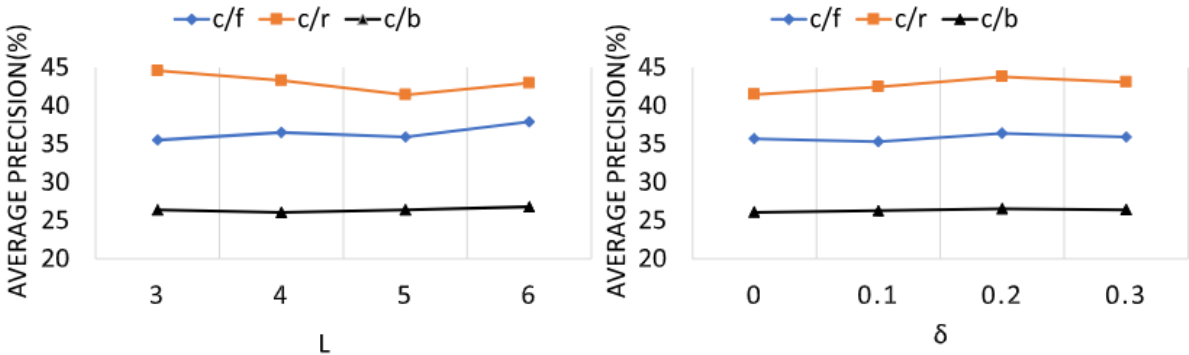
每个组件的有效性

表5 MLRT 中每个组件的消融结果 (%)。baseline：元学习框架的Faster RCNN。 RST：随机源转换模块。

GW：梯度加权模块

DG Setting	RST	GW	mAP
C to F	baseline		28.6
		✓	29.8
	✓		35.1
	✓	✓	36.4
C to R	baseline		42.1
		✓	43.2
	✓		43.2
	✓	✓	43.8
C to B	baseline		21.0
		✓	22.3
	✓		25.7
	✓	✓	26.5

参数灵敏度



$$w_i = \begin{cases} 1, & \cos \bar{\theta}_i - \cos \theta_i \leq \delta \\ \frac{\|\bar{g}_i\|_2}{\|g_i\|_2}, & \cos \bar{\theta}_i - \cos \theta_i > \delta \end{cases}$$

图21 超参数的消融结果。左图为不同批量大小L下的结果，右图为等式3中不同阈值delta下的结果。

3.1 实验结果与分析

消融实验

通过泛化系数分析域偏差

- 为了定量分析该问题，作者提出了一个新的指标：泛化系数G

$$G = \frac{2 \sum_n^N mAP_n}{1 + \sum_n^N \sum_{m, m \neq n}^N |mAP_n - mAP_m|} \tag{22}$$

表6 域偏差分析结果 (%)。 trget 1-3: 从CITYSCAPES (C)转变而来的虚假目标域。 最佳泛化系数以粗体突出显示

Method	Target	person	rider	car	truck	bus	train	motor	bike	mAP	G
Source-only	target1	23.7	31.6	41.3	13.2	33.5	15.3	16.0	26.0	25.1	2.5
	target2	16.1	20.5	34.1	9.2	19.5	2.8	12.1	22.5	17.1	
	target3	29.0	43.8	52.2	19.4	39.2	10.7	22.9	33.0	31.3	
MLDG	target1	31.3	41.8	53.3	26.0	40.7	8.5	29.1	29.6	32.5	11.0
	target2	34.8	46.4	53.7	30.8	43.5	17.6	31.6	35.6	36.8	
	target3	32.6	44.9	53.4	24.0	45.1	11.5	30.8	34.7	34.6	
FACT	target1	23.3	34.3	42.6	12.8	25.2	10.6	13.4	24.1	23.3	4.1
	target2	27.8	40.9	49.4	22.5	34.6	17.1	18.4	32.2	30.4	
	target3	31.6	45.5	52.7	25.0	45.7	4.0	27.9	35.4	33.5	
FSDR	target1	30.8	43.8	51.9	24.8	40.5	22.6	25.9	34.8	34.4	15.0
	target2	32.1	45.8	52.9	28.0	44.7	25.2	28.2	36.0	36.6	
	target3	35.2	48.7	53.3	27.3	46.7	17.3	31.5	40.0	37.5	
MLRT(Ours)	target1	37.2	50.5	53.5	27.9	50.5	19.4	33.6	40.7	39.2	46.3
	target2	37.2	51.7	53.5	27.6	52.6	16.9	32.8	41.0	39.2	
	target3	37.4	51.6	53.5	28.5	54.3	19.2	33.4	41.6	39.9	

3.1 实验结果与分析

消融实验

梯度不平衡分析

- 实验设置：在C、F、B这三个域中选择两个域来训练检测器 60,000 次迭代，在每次迭代中，按照之前提到的GW模块的算法，计算每个域的余弦相似度， $\cos\theta_1$ 和 $\cos\theta_2$ 。
- 两个余弦相似度之间的差 d 为：

$$d = |\cos\theta_2 - \cos\theta_1|. \quad (23)$$

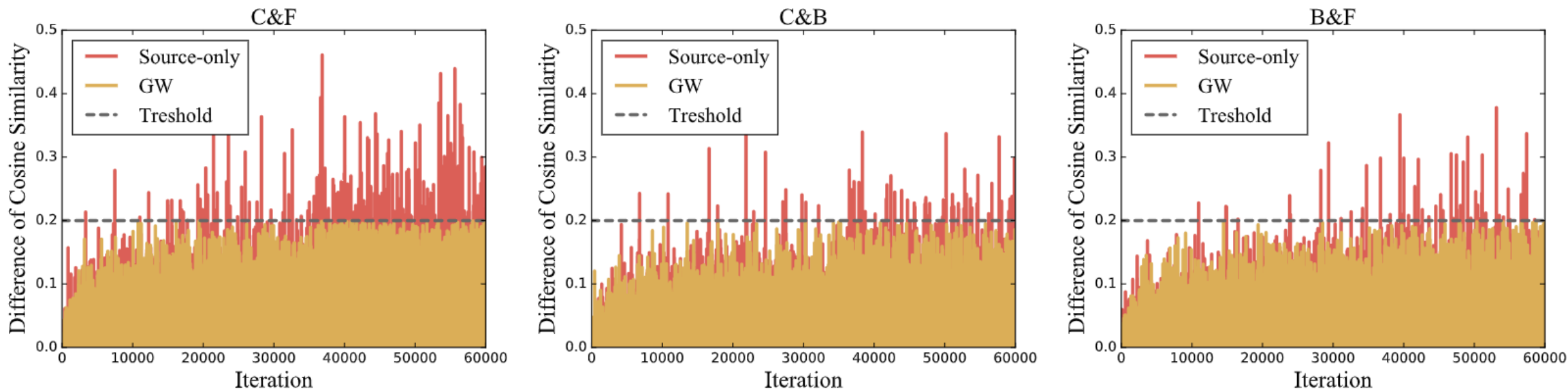


图22 带和不带GW模块的模型变体比较。余弦相似度差值越小，表示梯度越平衡。

4

总结

4 总结

随机变换元学习: Meta Learning on Randomized Transformations (MLRT)

- 首先作为域泛化模型, 能够解决更实际和更具挑战性的场景, 无需访问目标域且可以同时推广到广泛的未见过的目标域。
- 所提出的**随机谱变换(RST)**将源图像在**图像级别上**转换为各种新图像, 而无需访问模型参数和任何参考域。
- 所提出的域**梯度加权模块(GW)**是针对域之间的梯度平衡而定制的, 以进一步减轻**域偏差**。
- 部署了一种**元学习范式**, **理论分析**模型如何改善域偏差, 实现良好的泛化性。