OpenNet：面向开放环境罕见类别的自动驾驶检测系统

1 研究内容

本项目通过建立 OpenNet 端到端模型作为核心的自动驾驶检测系统提出针对开放世界 Corner Case 问题的解决方案，研究涉及的具体内容如下。

1.1 研究 Corner Case 问题的影响因素

假设训练数据的质量和模型训练的过程都可能会影响自动驾驶系统应对 Corner Case 的能力，本研究拟采取基于消融实验的分析方案，即如果在提升了训练数据的完备性和均衡性后模型的性能得到有效提升，则说明训练数据的质量欠佳确实可能导致 Corner Case 问题；同理，如果在模型训练过程中运用合理的技术手段改进模型的训练过程而使其性能得到提升，则说明模型的训练失当同样会导致 Corner Case 问题。如此，在分析导致 Corner Case 问题出现原因的同时，也可以有效开展实验研究。

本研究拟从数据和模型两个角度出发，寻找提升模型应对 Corner Case 能力的方法。首先，分别从两个角度阐述现有样本数据和模型训练过程中可能存在的问题，即本研究为什么选择从这两个角度开展研究：

从数据的角度。自动驾驶领域的数据集是符合长尾分布的，且质量可能参差不齐：从类别上说，某些常见类（Common classes）的样本数量（汽车，行人，红绿灯等）会远远高于稀有类（婴儿车，宠物，新型路障等）；从场景来说，暴雨暴雪、雾霾、深夜等场景的样本数量会远远少于普通场景（例如，CODA 数据集中有 75% 的图像是晴天，4% 的雨天，9% 的夜晚）。因此，自动驾驶数据集所拥有的天然的，同时也极具挑战性难题的就是长尾效应，这不仅会导致模型对已知类的预测偏倚情况，更可能阻碍模型对新类别等稀有样本（corner cases）的学习和检测，存在潜在的安全隐患。

从模型的角度。在开放的现实世界中，自动驾驶汽车必须具备与外部环境进行有效交互的能力，这其中的难点包括：鉴别从未见过的新类别，道路场景的风格和内容比较罕见，模型以较低的置信度输出识别结果等。上述问题可能会导致模型在训练过程中虽然表现良好，但是在实际测试时性能大幅下滑（例如在 CODA 数据集上的测试结果）的问题，导致自动驾驶技术无法走向更高的级别。

由上述分析，本研究对 Corner case 问题进行初步定义与描述：在自动驾驶的开放世界场景中，凡可能影响模型对道路目标进行有效检测与识别，从而导致潜在风险的因素，都应属于 Corner case 问题的研究范畴。

1.2 基于 Corner Case 的影响因素建立端到端模型为核心的自动驾驶检测系统

本研究拟提出的开放世界检测模型训练方法主要包含 Corner case 样本生成模块和 Corner case 模型泛化能力提升两个模块。

在 Corner case 样本生成中，本研究主要通过 GAN 的方法生成更多的稀有类训练样本和对抗样本，探索训练数据是否导致了 Corner Case 问题的发生并进行有效解决。a、研究基于风格迁移的样本生成方法，灵活控制生成模型对道路环境场景的渲染，解决不同道路风格场景的样本数量差异导致的长尾效应问题；b、研究基于语义解耦和特征生成的样本生成方法，能够自由地生成将样本主体放置于各种情景中的高质量新样本，解决不同主体的样本数量差异导致的长尾效应问题；c、研究基于对抗攻击策略的对抗样本生成方法，提升模型对极端样本的鲁棒性。

在 Corner case 模型检测中，旨在提升模型的 OOD（out of distribution）泛化能力，即模型在面对开放世界中未知类别、未知场景和突发情况时，能够保持良好的检测性能，做出高可信的决策，降低风险发生的概率。a、不确定性方法研究，度量模型的不确定性属性，追溯模型的决策缘由，保障在实际使用时面对异常能及时做出处理；b、域适应方法研究，减少训练和真实环境的领域风格差距与内容差距，限制域偏移的程度；c、开集识别方法研究，增强类内紧凑性和类间分离性，提高模型对已知类和未知类的鉴别能力，提升识别精度；d、增量学习方法研究，使模型具备对新类别增量学习的能力，以最符合人类认知规律的方式提升模型的自适应能力。

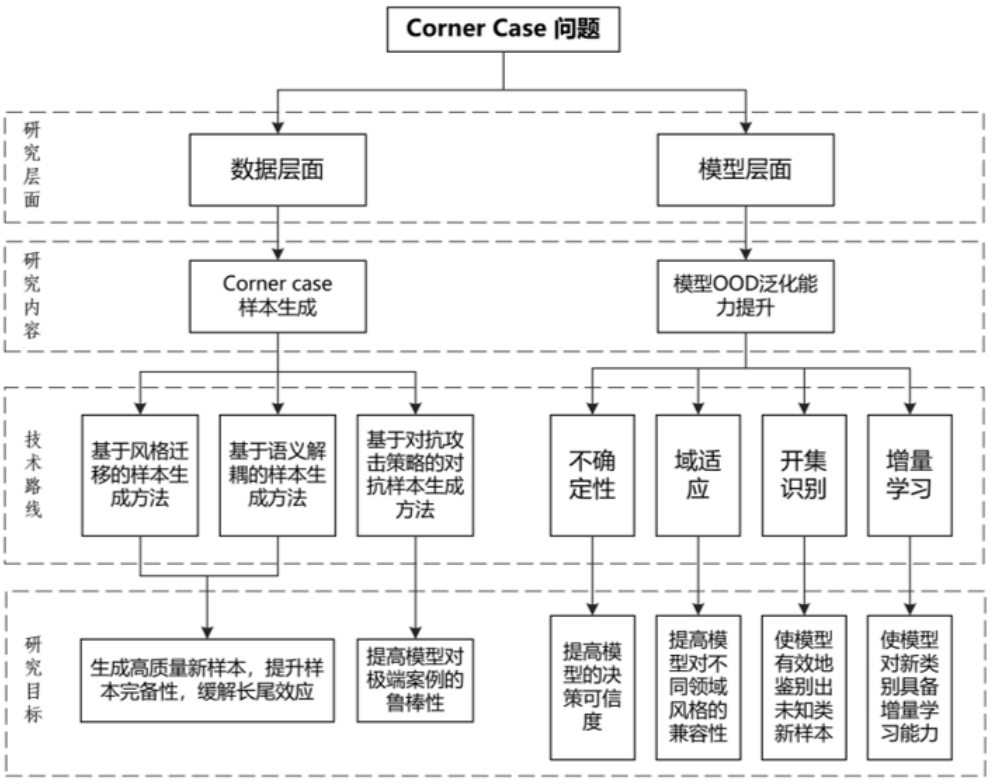

图 1: 研究框架

1.3 拟解决的关键问题

1. 对于自动驾驶 Corner Case 问题目前尚未有一个公认的定义、术语和描述。Corner Case 可以是对于数据而言的，也可以是对于模型而言的。在实际场景中，明确 Corner Case 的出现是由于训练数据导致的还是由于模型训练过程导致的；

2. 自动驾驶场景中的罕见类别与场景样本少且种类繁多，如何利用已有的训练数据训练模型，使模型在测试阶段能够处理各种罕见的类别与场景。

2 开放世界的检测方法

研究内容为 Corner Case 问题中的关键场景——新类别。广义上新类别指异常的空间或者时间集块，抑或是已知过程分布的变化；狭义上，新类别指场景不属于任何自动驾驶基准数据集的类别，抑或属于其中自动驾驶基准数据集类别的新实例。在现实世界的自动驾驶场景里，车辆行驶在开放的环境中，并会与在训练过程中未遇到的周围物体进行交互。因此，识别并定位自动驾驶场景中的未知类别是一项具有挑战性但又关键的任务。

项目拟使用基于 Towards Open World Object Detection（OWOD）的开放世界检测技术作为本项目的方法。

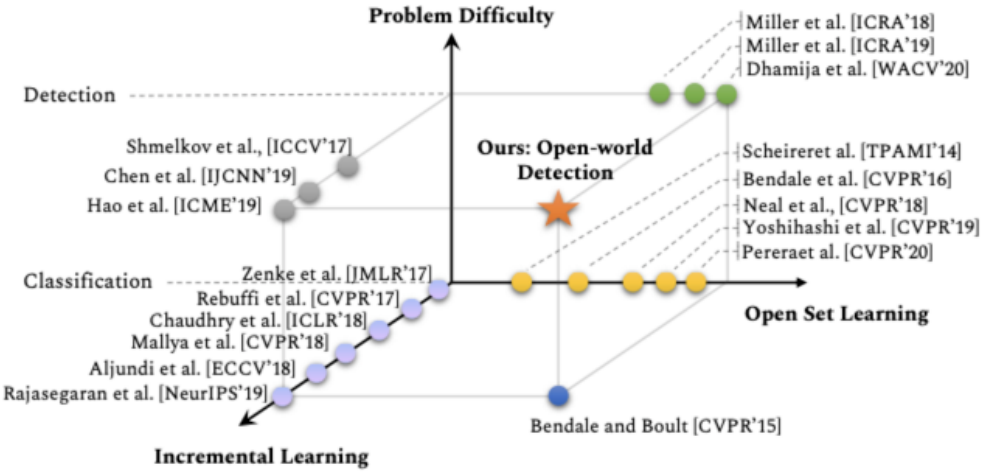

图 2: 开放世界的定义

开放世界（Open World）是指存在未知类别、未知实例的情况下，系统需要能够根据当前场景进行确定性决策的环境。开放世界目标检测就是在开放世界中，对于未知类别的目标进行检测和识别。并增量学习未知类。

在传统的目标检测中，模型会在已知的类别中进行训练，然后在测试时对已知类别和未知类别进行分类。但在现实场景中，我们无法预知所有可能出现的目标类别。因此，开放世界目标检测需要能够识别和处理未知类别，以便使系统能够在未知类别出现时做出正确的行动。

为了解决这个问题，OpenNet 项目使用了一种基于 Towards Open World Object Detection（OWOD）的开放世界检测技术。该技术可以识别和定位未知类别的目标，从而提高自动驾驶系统的鲁棒性和安全性。

3 Open World Object Detection 定义

在 t 时刻，已知的目标类别为：

$$K^t = 1, 2, ..., C \in \mathbb{N}^+ \tag{1}$$

未知类别为：

$$U = C + 1, ... \tag{2}$$

K 的训练集为：

$$D^t = X^t, Y^t \tag{3}$$

$$X = \{I_1, ..., I_m\} \tag{4}$$

$$Y = \{Y_1, ..., Y_M\} \tag{5}$$

X, Y 分别为图片和标注信息，其中每张图片 Y_i 包含了多个目标实例，每个实例都有其标签和未知信息。

$$Y_i = \{y_1, y_2, ..., y_k\} \tag{6}$$

在 OWOD 的设定中，模型 M_c 用于检测所有的 C 个已知类别，同时可以将未知的目标标记为 0，未知的实例集合 U_t 筛选出 n 个新类别并分为为该类别准备相应的样本标签对，通过增量学习的方式学习模型 M_{C+n} .

已知类别更新为：

$$K_{t+1} = K_t + \{C + 1, ..., C + n\} \tag{7}$$

模型可以持续循环该过程进行开放世界的自动驾驶目标检测。

4 创新点

1. 多尺度的特征提取器

2. 全局特征的 proposal 计算

5 模型架构

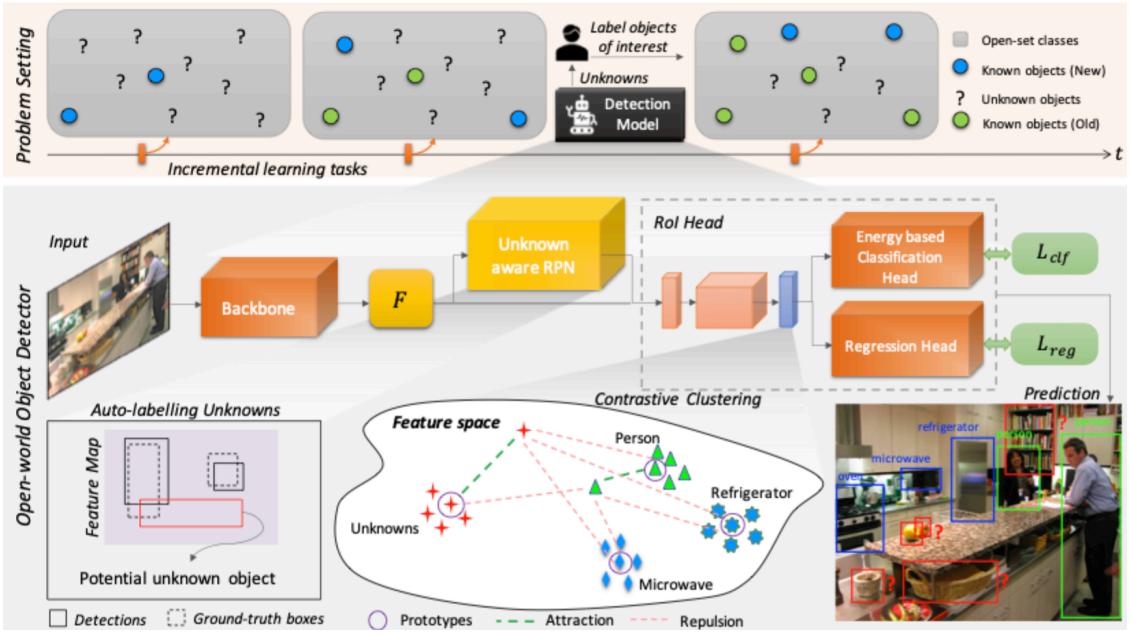

图 3: 模型架构

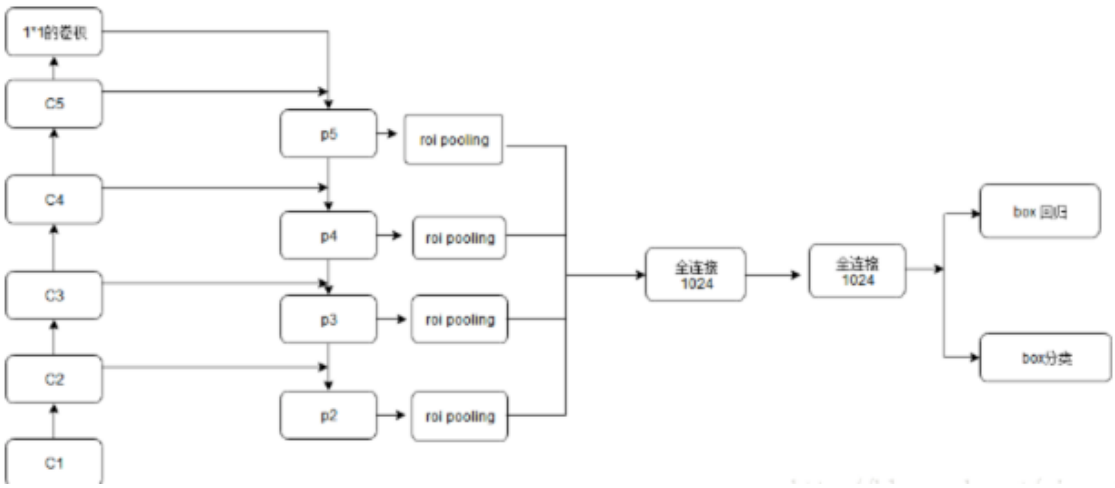

图 4: Backbone 架构

6 Pipeline

- Backbone 提取图像的特征 F .
- F 分别输入给 ROI Head 和 Unknown aware RPN.
- Unknown aware RPN 会产生候选框，区分前景和背景，背景框为没有被标注的区域，这些背景

框中分数较高的可能为 unknown 的实例，选其中 top_k 个候选框为未知类别目标。

- ROI Head 中经过全连接层生成 cls 对应的 feature f_c .
- 根据 F_{store} 更新每个类的类原型 P

$$P = \{p_0 ... p_c\} \tag{8}$$

- 使用类原型进行对比聚类，将聚类损失加入到分类损失和回归损失中作为额外信息 f_c 输入 Energy based Classification Head 和 Regression Head 分别做实例分类和实例 Bounding Box 的回归形成最终的图片 Prediction.

7 多尺度的特征提取器（FPN）

- 目标检测中最具挑战性的问题就是目标的尺度变化问题（scale variance）。在目标检测中，物体的形状和尺寸大小不一，甚至可能出现一些极小、极大或者极端形状（如细长型、窄高型等）的物体，这就给目标的准确识别和精准定位带来了极大困难。
- 现有的针对目标尺寸变化问题而提出的算法中，较为有效的算法主要有图像金字塔和特征金字塔，这两者的共同思想就是利用多尺度特征来检测不同尺寸的物体。
  - 图像金字塔：
    - 允许任意大小的图像输入，但是时间损耗巨大，且会占据大量的存储空间，每个不同分别率的输入图像都要通过同一个 CNN，这个过程存在大量的冗余计算。
  - 特征金字塔：
    - 为减少冗余计算，改进为多尺度的特征输入，即输入单一分辨率的图像，经过卷积得到不同分辨率的特征图。
- 多尺度特征融合 + 多尺度特征预测
  - FPN 通过引入多尺度的特征融合，并进行多尺度的特征预测，减小了目标尺寸变化的影响，大大提高了目标检测的准确度，在相邻特征层融合时，FPN 采用的上采样方式是双线性插值，特征叠加方式为逐元素相加。

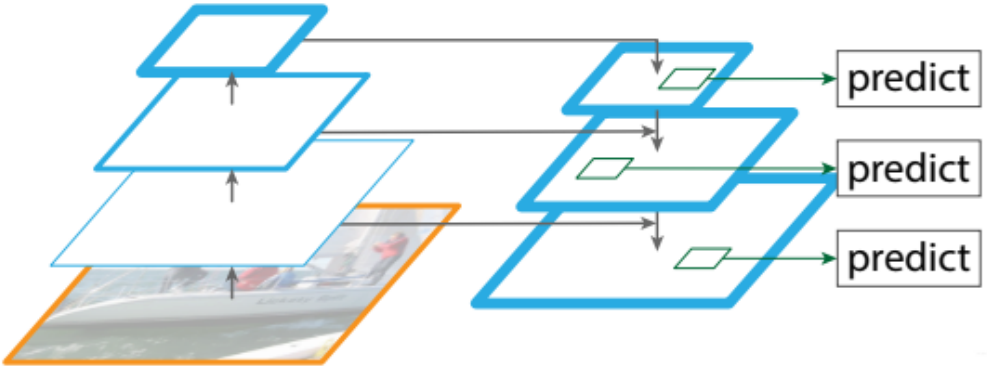

图 5: 示例图

8 Feature Store

对每个类维护一个全局 $F_c^{'}$.

$$F_c = \frac{1}{n}\sum_{i=1}^{n} f_{ci} = \frac{1}{n} * (F_c^{'} * (n-1) + f_c) \tag{9}$$

考虑到内存的限制无法保存所有的 feature，因此通过动态更新 $F_c^{'}$ 实现近似全局平均 F_c ，同时维护一个 k 长度的队列保存后 k 个 f_c ，F_{store} 保存该队列和 F_c .

9 Auto-Labeling Unknowns with RPN

特征图经过 RPN 后会根据特征图生成多个实例的 Proposals，Proposals 由 *anchor* 的偏移和 Foreground probability 组成。

从背景 *Proposals* (first stage detection) 里面选择 top-k 最高 *objectness scores* 的 *candidates* 即为未知物体的 *candidate*.

在对比聚类中，未知类别也有其对应的原型向量 p_0 .

将预测框中 objectness 的 topK 直接归类为未知目标，将其特征加入未知类的 feature 队列 q_0 中。

10 对比聚类

针对隐空间中的特征表示，相同的类别应该距离相近，不同的类别距离应该较远；基于这一期望，为骨干网络抽取的特征添加一个聚类约束。

10.1 具体实现

对每个类别的隐空间特征，统计一段时间内迭代样本的特征均值，作为聚类中心，并约束期望该类样本特征都靠近该中心，其他类的样本特征远离该中心；聚类中心在训练过程中不断更新。

对比损失：

$$L_{cont}(f_c) = \sum_{i=0}^{C} l(f_c, p_i) \tag{10}$$

$$l(f_c, p_i) = \begin{cases} D(f_c, p_i), & \mathrm{i} = c \\ max\{0, \Delta - D(f_c, p_i)\}, & otherwise \end{cases} \tag{11}$$

(11) 中 D 为距离函数，Δ 为相似阈值，不同类别实例间的距离要大于该阈值。

在训练时，通过最小化对比损失来保证特征空间上的类别分割。

Algorithm 1 Algorithm COMPUTECLUSTERINGLOSS

Input: Input feature for which loss is computed: $\boldsymbol{f}_c$; Feature store: $\mathcal{F}_{store}$; Current iteration: i; Class prototypes: $\mathcal{P} = \{\boldsymbol{p}_0 \cdots \boldsymbol{p}_C\}$; Momentum parameter: η.

Initialise $\mathcal{P}$ if it is the first iteration.
$\mathcal{L}_{cont} \leftarrow 0$
if $i == I_b$ **then**
    $\mathcal{P} \leftarrow$ class-wise mean of items in $\mathcal{F}_{Store}$.
    $\mathcal{L}_{cont} \leftarrow$ Compute using $\boldsymbol{f}_c$, $\mathcal{P}$ and Eqn. 1.
else if $i > I_b$ **then**
    if $i\%I_p == 0$ **then**
        $\mathcal{P}_{new} \leftarrow$ class-wise mean of items in $\mathcal{F}_{Store}$.
        $\mathcal{P} \leftarrow \eta\mathcal{P} + (1-\eta)\mathcal{P}_{new}$
    $\mathcal{L}_{cont} \leftarrow$ Compute using $\boldsymbol{f}_c$, $\mathcal{P}$ and Eqn. 1.
return $\mathcal{L}_{cont}$

图 6: 算法 1

对比损失的计算过程如图 5 所示，为了保证原型向量有相对的准确性，仅当超过一定迭代次数之后才开始计算损失值，之后每次迭代 $I_b * n$ 次就以动量的形式更新一次原型向量。这样可以避免原型向量变化过大的问题，得到的损失值添加到检测损失值中进行端到端的学习。

10.2 优势

1. 可以帮助网络辨别 unknown 类别的物体与已知类别的表示有何不同；

2. 促进网络在不覆盖潜在空间中原有类别表示情况下学习未知类别的潜在表示。

11 Energy Based Unknown Identifier

由于 Opern World Detection 场景包含未知类别的特性，传统的 softmax 分类器可能会给出不可控的结果，所以采用基于能量的分类器 (EBM)，能够学习输入特征与标签之间的匹配程度，用来识别未知目标。

给定特征 $f \in F$ 与标签 $l \in L$, 学习一个能量函数 $E(F, L)$，能够通过 $E(f) : R^d \to R$ 得到一个能用于描述特征与标签之间的匹配程度的标量 (即能量)。这里，论文采用了 Helmholtz free energy 公式计算所有标签的结果之和：

$$E(f) = -Tlog \int_{l'} exp(-\frac{E(f, l')}{T}) \tag{12}$$

(12) 中 T 是温度参数。通过 Gibbs 分布，可以将各标签的能量转化成类似 softmax 那样的效果：

$$p(l|f) = \frac{exp(\frac{g_l(f)}{T})}{\sum_{i=1}^{C} exp(\frac{g_i(f)}{T})} = \frac{exp(-\frac{E(f,l)}{T})}{exp(-\frac{E(f)}{T})} \tag{13}$$

综上，用于分类模型的 free energy 公式：

$$E(f; g) = -Tlog \sum_{i=1}^{C} exp(\frac{g_i(f)}{T}) \tag{14}$$

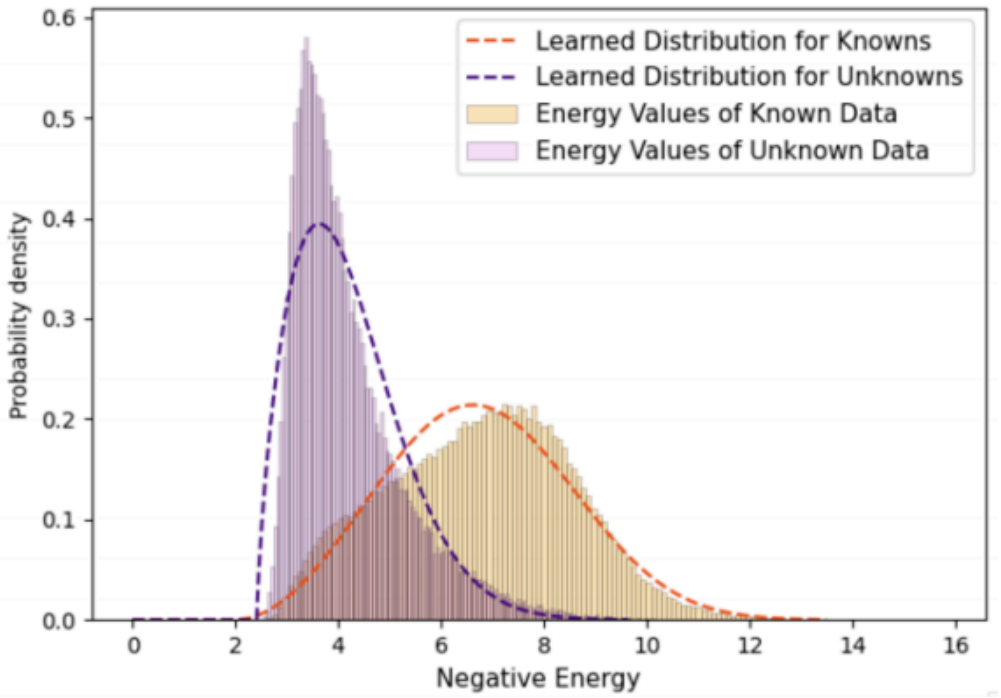

图 7: 能量值

由于 ORE 用了对比聚类对特征进行分割，已知类别的能量值和未知类别的能量值也有明显的差别。对已知类别和未知类别的能量值分布进行 shifted Weibull distributions 建模，得到 $\xi_{kn}(f)$ 和 $\xi_{unk}(f)$，当 $\xi_{kn}(f) \leq \xi_{unk}(f)$ 时，可以认为该目标属于未知类别。

12 Alleviating Forgetting

在对识别出来的未知目标进行标注后，得到了新的数据集，如果将所有数据集混合重新训练会很耗时且不够灵活，所以只能使用新数据集进行增量学习，这就需要解决新类别训练对旧类别识别效果的影响。

论文参照了增量学习的 SOTA 方法，使用简单的样本回放策略来保证旧类别的效果，先构造一个小的样本集 (exemplar set)，包含每个类别的 N_{ex} 个样本，每次使用全量新数据集进行增量学习后，都使用小样本集进行一次 finetune 训练，这样就能很好地保证旧类别的效果而且不耗时。

13 Experiment and Results

我们使用 SODA10M[1] 和 CODA[2] 作为训练和验证集，二者皆是自动驾驶领域的数据集。前者含有更多的基础类（'pedestrian', 'cyclist', 'car', 'truck', 'tram', 'tricycle'），CODA 则包含更多的 Novel 类（'bus', 'bicycle', 'moped', 'motorcycle', 'stroller', 'cart', 'construction_vehicle', 'dog', 'barrier', 'bollard', 'sentry_box', 'traffic_cone', 'traffic_island', 'traffic_light', 'traffic_sign', 'debris', 'suitcace', 'dustbin', 'concrete_block', 'machinery', 'garbage', 'plastic_bag', 'stone'）.

因此首先在 SODA10M 上进行预训练，再在 CODA 上进行 finetune.

表 1: use SODA10M first, then use CODA to finetune

	Task1	Task2
Semantic split	SODA10M	CODA
training image	4967	8792
test image	490	870
train instances	41110	26318
test instances	4100	2500

表 2: compare OWOD with OpenNet

	CODA							
	clsAccuracy	falseNegative	fgclsAccuracy	lossBoxReg	lossCls	lossRpnCls	lossRpnLoc	totalLoss
OWOD	0.9747	0.1288	0.8532	0.2795	0.0877	0.0072	0.0371	0.4278
OpenNet	**0.9803**	0.1422	**0.8324**	**0.2197**	**0.0621**	0.0160	**0.0233**	**0.3204**

实验验证了多尺度特征提取融合和全局特征原型的优越性。

14 Implementation Details

使用 2 张 1080Ti 进行并行训练，最大 iteration 为 10000，batchsize 设置为 12，lr 为 0.01，其他超参数与原论文一致。

15 References

[1] Joseph, K. J., Khan, S., Khan, F. S., Balasubramanian, V. N. (n.d.). Towards Open World Object Detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, 5830–5840.

[2] Gupta, A., Narayan, S. (2022). OW-DETR: Open-World Detection Transformer. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 9235–9244.

[3] Li, L. H., Zhang, P. (2022). Grounded Language-Image Pre-Training. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 10965–10975.

[4] Gu, X., Lin, T. (2022). Open-Vocabulary Object Detection via Vision and Language Knowledge Distillation. ICLR.

[5] Yoshihashi, R., Shao, W. (2019). Classification-Reconstruction Learning for Open-Set Recognition. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 4016–4025.

[6] Ganin, Y., Lempitsky, V. (2015). Unsupervised Domain Adaptation by Backpropagation. Proceedings of the 32nd International Conference on Machine Learning, 1180–1189.

[7] Yilmaz, I. (2020). Practical Fast Gradient Sign Attack against Mammographic Image Classifier.

[8] Zhong, Z., Cui, J. (2021). Improving Calibration for Long-Tailed Recognition. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 16489–16498.

[9] Mirza, M., Osindero, S. (2014). Conditional Generative Adversarial Nets.

16 Supplement

16.1 OpenNet 效果图

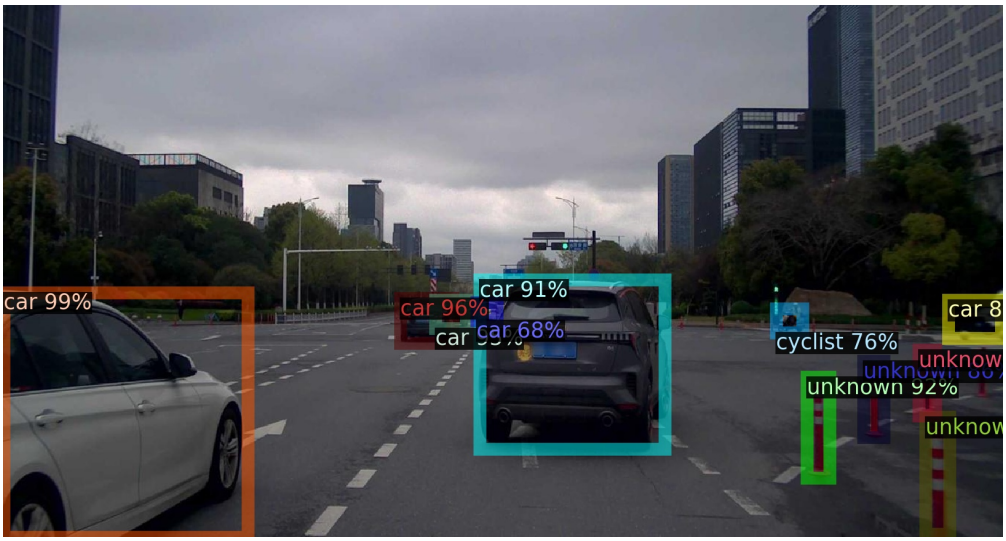

图 8: 效果图 1

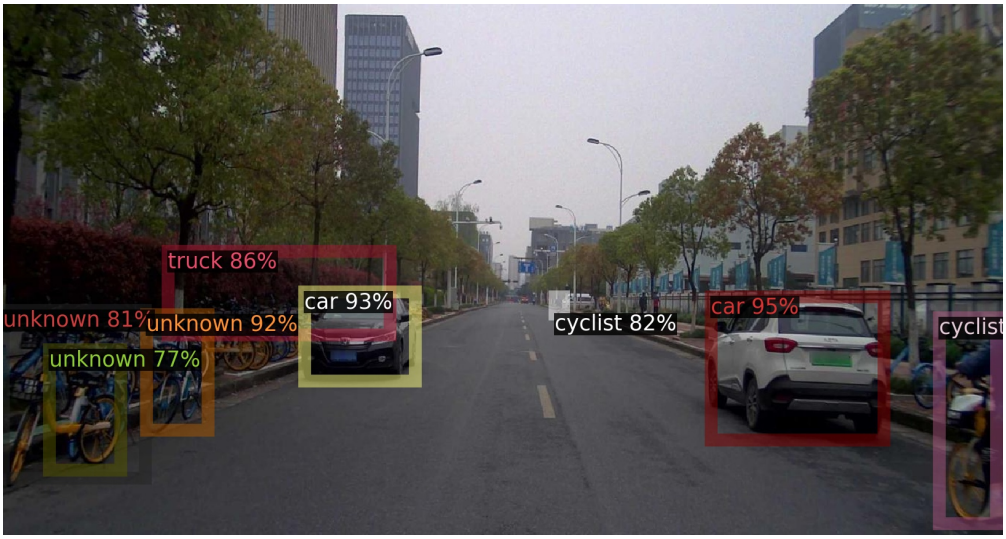

图 9: 效果图 2

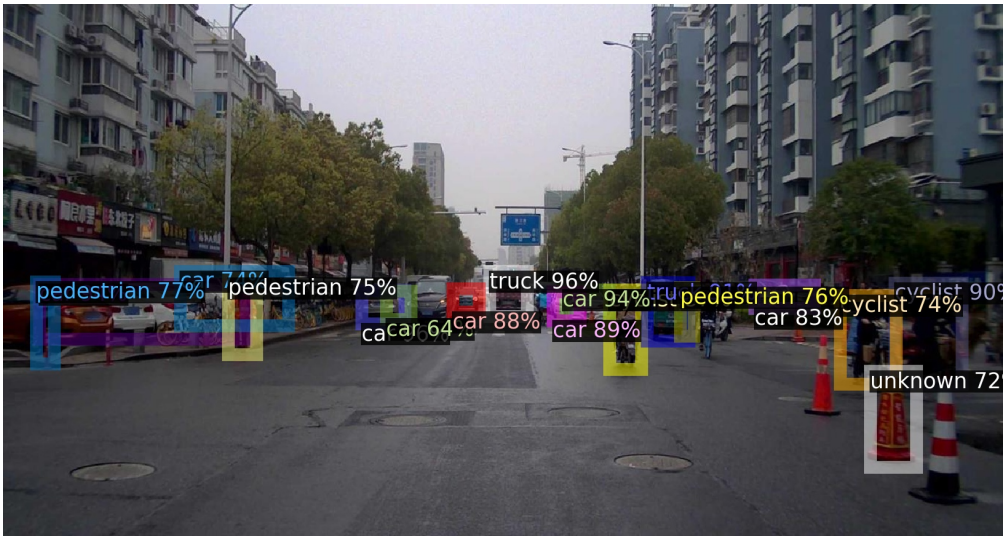
pedestrian 77%
pedestrian 75%
truck 96%
car 94%
pedestrian 76%
car 83%
cyclist 74%
car 88%
car 89%
unknown

图 10: 效果图 3

16.2 OpenNet 实验过程图

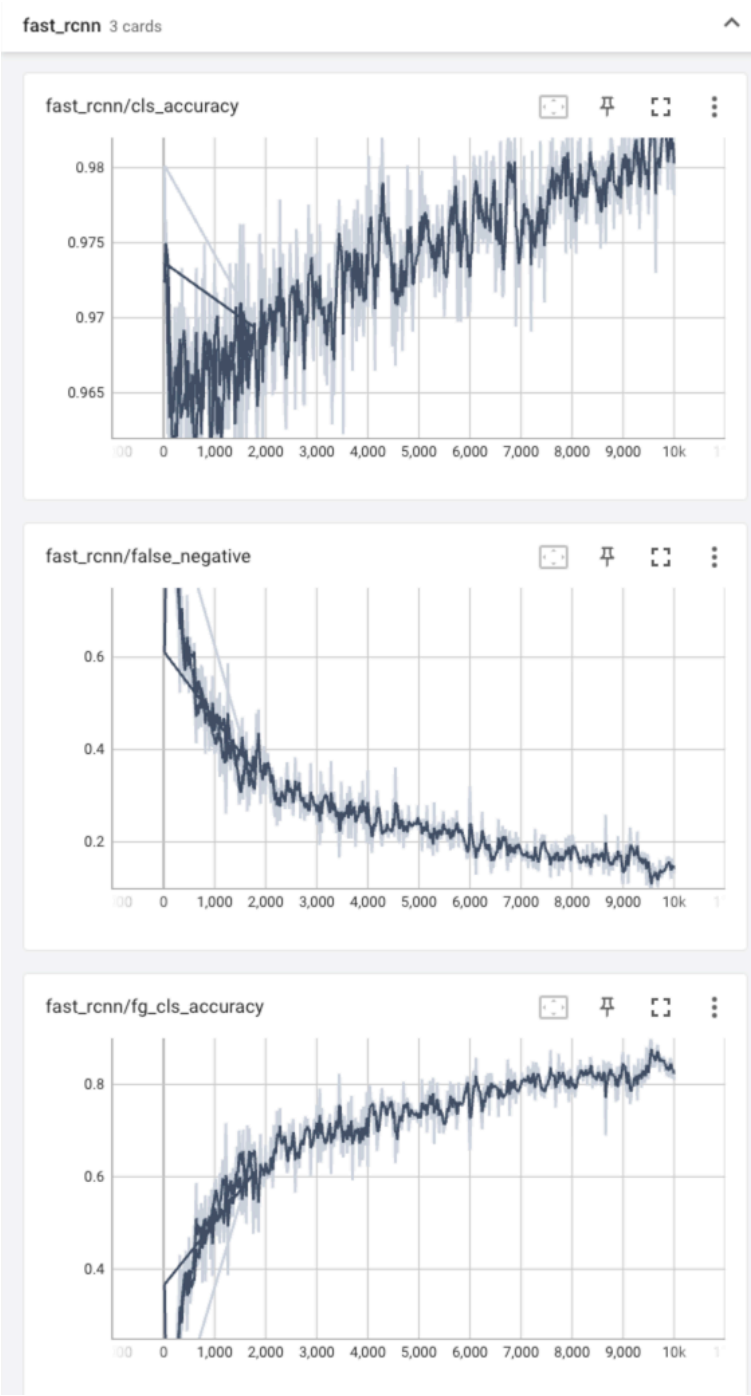

图 11: 实验过程图 1

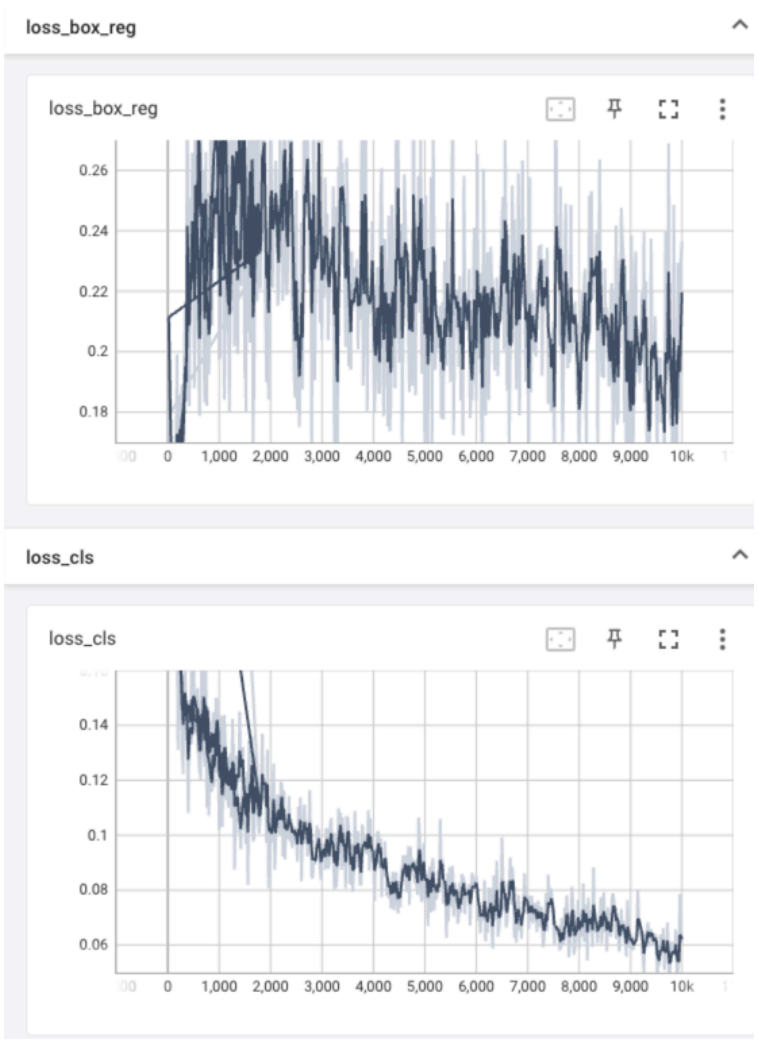
loss_box_reg
loss_box_reg
0.26
0.24
0.22
0.2
0.18
0
1,000
2,000
3,000
4,000
5,000
6,000
7,000
8,000
9,000
10k
loss_cls
loss_cls
0.14
0.12
0.1
0.08
0.06
0
1,000
2,000
3,000
4,000
5,000
6,000
7,000
8,000
9,000
10k

图 12: 实验过程图 2

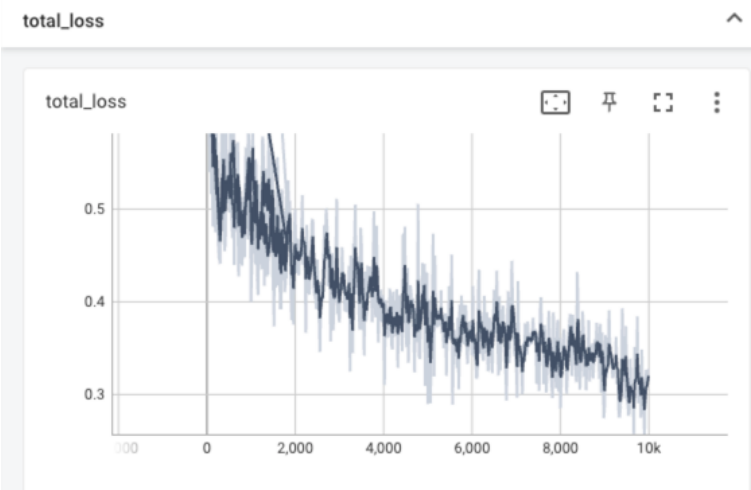

图 13: 实验过程图 3

16.3 OWOD 训练过程图

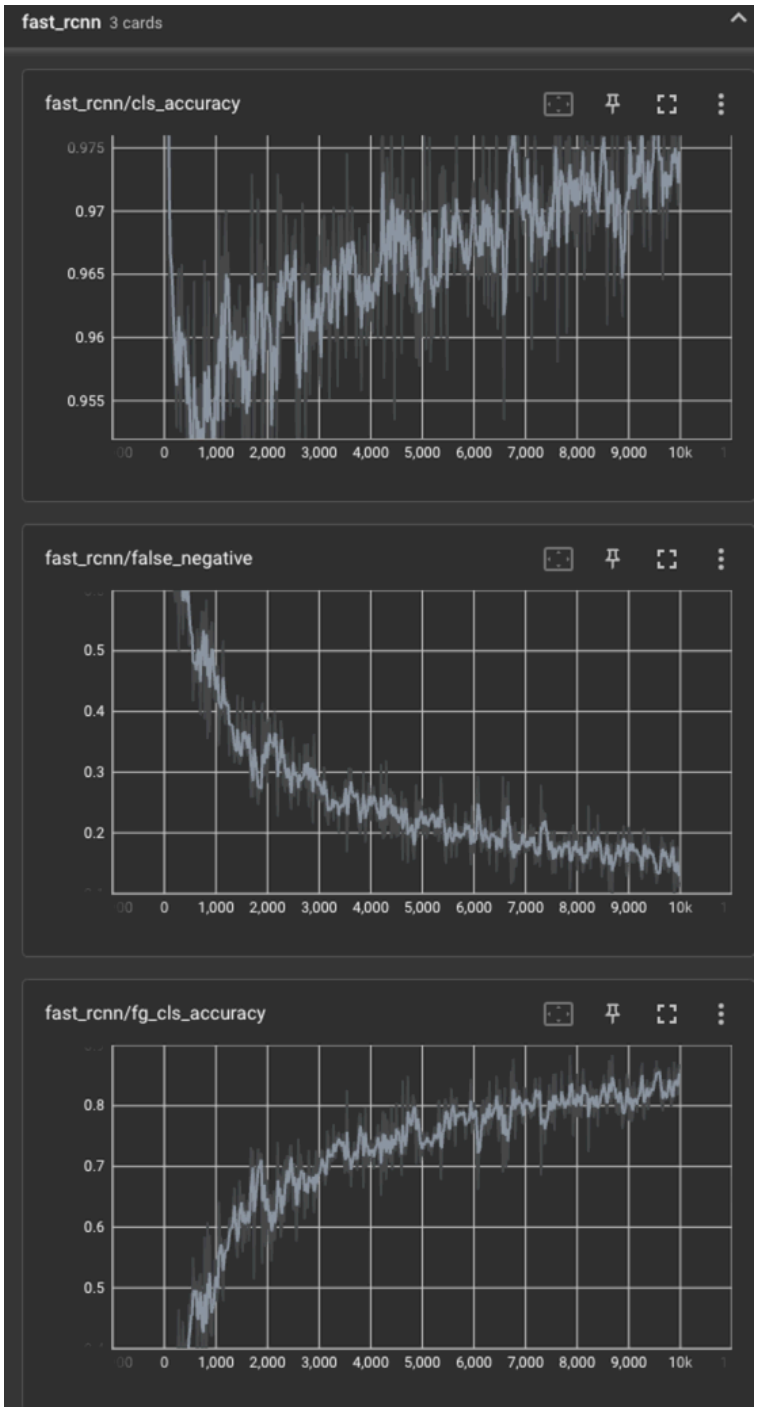

图 14: 训练过程图 1

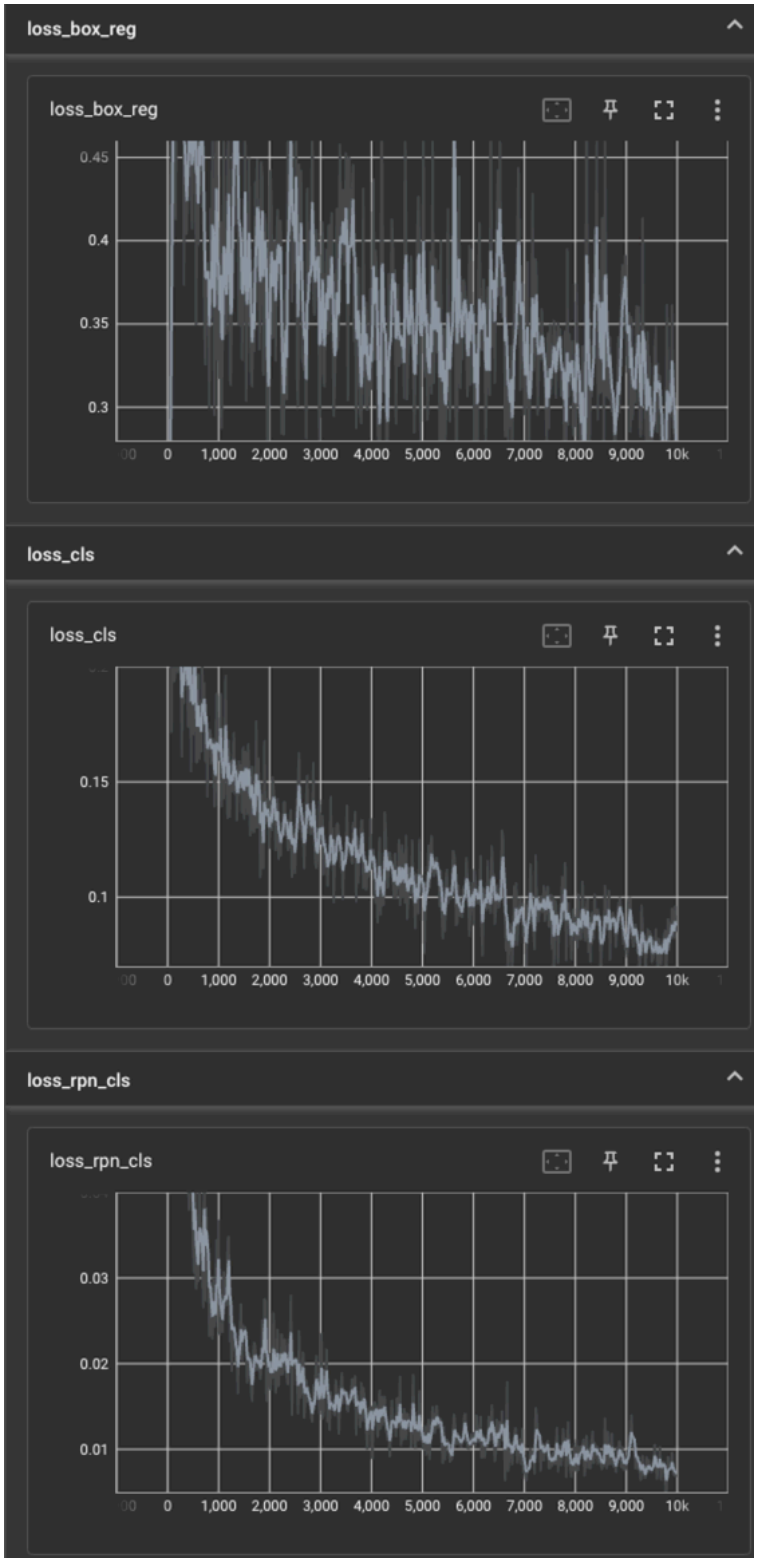
loss_box_reg
loss_box_reg
0.45
0.4
0.35
0.3
0
1,000
2,000
3,000
4,000
5,000
6,000
7,000
8,000
9,000
10k
loss_cls
loss_cls
0.15
0.1
0
1,000
2,000
3,000
4,000
5,000
6,000
7,000
8,000
9,000
10k
loss_rpn_cls
loss_rpn_cls
0.03
0.02
0.01
0
1,000
2,000
3,000
4,000
5,000
6,000
7,000
8,000
9,000
10k

图 15: 训练过程图 2

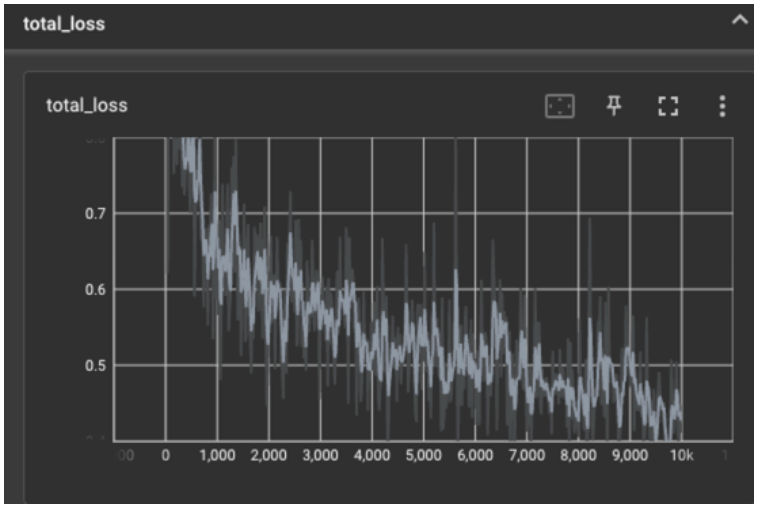
total_loss
total_loss
0.7
0.6
0.5
0
1,000
2,000
3,000
4,000
5,000
6,000
7,000
8,000
9,000
10k

图 16: 训练过程图 3