

自动驾驶与智能交通领域中 协同感知算法研究与实践

北方工业大学

葛钰峒

更新: July 9, 2023



目录

1	个人介绍	3			
2	协同感知领域相关技术背景介绍	4			
2.1	机遇与挑战	7			
2.1.1	可靠的通讯	7			
2.1.2	跨模态技术的应用	7			
2.1.3	大规模数据集	7			
3	论文概述	8			
3.1	学生模型 (中期协作)	9			
3.2	教师模型 (早期协作)	12			
3.3	知识蒸馏	13			
3.3.1	概述	13			
3.3.2	本文方法	14			
3.4	实验设置	15			
3.4.1	定量评估	15			
3.4.2	定性评估	17			
3.5	消融实验	18			
4	代码实践	19			
4.1	实验环境	19			
4.2	实验步骤	19			
4.3	模型训练	20			
4.4	测试结果	21			
4.5	样例抽取	22			
5	算法优化	23			
5.1	思路一: 特征融合	23			
5.1.1	遇到的问题	24			
5.1.2	Loss 曲线对比	25			
5.1.3	思路一总结	26			

1 个人介绍

姓名: 葛钰骁

学校: 北方工业大学

专业: 计算机科学与技术 (创新实验班)

邮箱: yuyao.ge.work@gmail.com

绩点及排名: 93.39 3/28

学科竞赛: 天梯赛个人国三, 蓝桥杯国三



科研成果:

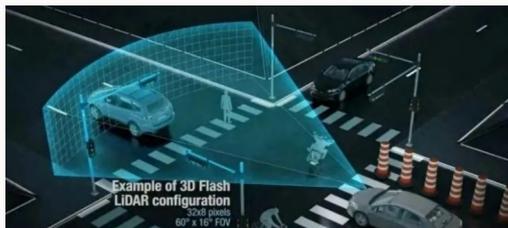
- 2022-2023 Attack based on data:A novel perspective to attack sensitive points directly | CCF-C/SCI 四区 | 一作, 录用 | 深度学习, 对抗攻击, 进化算法, 时序数据
- 2021 Vision Transformer Based on Knowledge Distillation in TCM Image Classification | CCET,EI 检索 | 一作, 见刊 | 视觉 Transformer, 知识蒸馏, 深度学习
- 2022 一种车牌识别方法、装置、电子设备及存储介质 | 国家发明专利, 第一发明人
- 2023 面向游戏与黑白漫画的非真实感实时渲染框架 | 中文核心,CCF T2 类 | 四作, 录用 | 写作指导
- 2023 Unsupervised one-point attack based on differential evolution | AANN,EI 检索 | 三作, 录用 | 深度学习, 对抗攻击, 进化算法
- 2023 基于数据特征的无监督对抗攻击 | 中文核心,CCF T2 类 | 中文核心一作, 在审 | 深度学习, 对抗攻击

2 协同感知领域相关技术背景介绍

感知是自动驾驶系统的主要任务之一,然而单智体有限的感知和计算能力、高昂的成本导致单车感知能力达到瓶颈。为了解决这个问题,协同感知技术应运而生。通过协同感知技术,单个车辆可以与附近智体共享信息,使其突破自身位置的限制,感知到被遮挡或远距离环境的情况。

协同感知算法根据环境传感器类型可分为两类,即基于视觉 (Vision-based) 的算法和基于点云 (Point-cloud-based) 的算法。

1. **基于视觉 (Vision-based) 的算法:** 基于相机传感器的数据的感知方法通过分析或人工智能技术剖析图像像素以检测环境中的车辆、行人以及其他障碍物。基于视觉的算法使用几何、光流、颜色或者其他图像特征进行检测。
2. **基于点云 (Point-cloud-based) 的算法:** 基于点云的感知主要依赖于 LiDAR 传感器输出的对空间描述的 3D 点云数据。算法可以通过点的密度、几何形状或图案从大量点云数据中检测出物体。在自动驾驶领域,3D 点云数据一般通过 LiDAR(激光雷达) 传感器进行扫描采集获取, LiDAR 获取方式据有高精度、高密度以及快速等特点,也可以通过双目视觉图像进行三维重建获取



根据协作方式, 有三种主流协作手段, 如下图1所示, 分别为早期协作, 中期协作, 晚期协作。

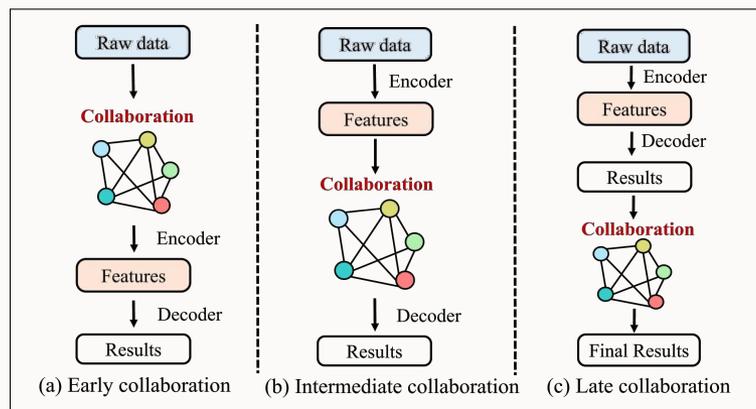
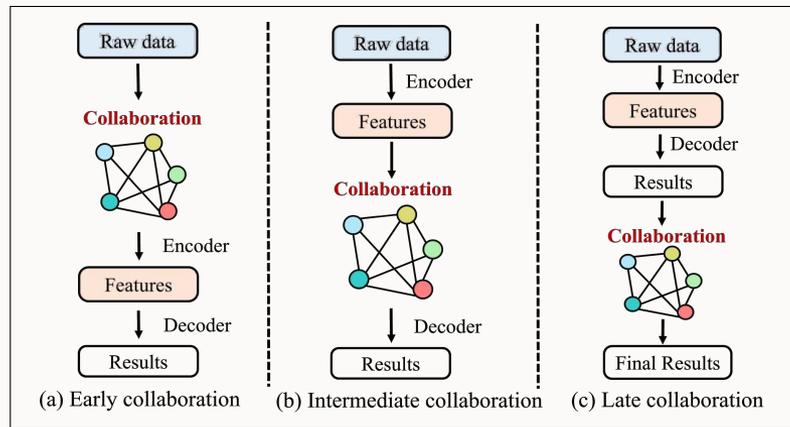


图 1: Three collaboration modes

1. **早期协作:** 智体在**输入空间**中协作, 共享**原始感官数据**。通过该方法协作的智体可以汇总所有智体的原始测量值, 即拥有全局视角。可以从根本上解决遮挡和长距离的问题, 但是这种方法对带宽的要求大, 容易因网络负载过大而拥塞。
2. **晚期协作:** 智体在**输出空间**中协作。将自身的感知结果与其他智体的感知结果融合。但该方法误差大, 噪声多。
3. **中期协作:** 智体在**特征空间**中协作。智体可以将自身的特征与其他智体的特征进行融合, 然后使用解码器产生感知结果。与早期协作相比, **更节省带宽**; 与晚期协作相比, **感知更为准确**。在实践中, 对中期协作策略的优化主要有两个方面: (i) 如何从原始数据中提取有效的 Feature。 (ii) 如何最大程度地融合其他智体的特征来提升本智体的感知能力。



除此之外, 还可以结合以上多种方法的协作手段, 即混合协作。例如, Arnold E, et al¹提出在高可见的环境下使用后期协作策略, 在低可见的环境中使用早期协作策略。此外, Li Y, et al²中提出的 DiscoNet 使用一个利用了前期协作策略的**教师模型**来指导使用了中期协作策略的**学生模型**。而在推理阶段, 则不再使用教师模型, 这种方法可以在低带宽的环境下保证卓越的性能。这也是我在后面主要介绍的方法。

¹Arnold E, Dianati M, de Temple R, et al. Cooperative perception for 3D object detection in driving scenarios using infrastructure sensors[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 23(3): 1852-1864.

²Li Y, Ren S, Wu P, et al. Learning distilled collaboration graph for multi-agent perception[J]. Advances in Neural Information Processing Systems, 2021, 34: 29541-29552.

2.1 机遇与挑战

2.1.1 可靠的通讯

高效的协作依赖于可靠的通讯,为此**通信的鲁棒性**显得尤为重要。然而在实际应用中,通信仍面临各种问题。

- 受**带宽的限制**,当车辆数量增大时,网络会拥塞。
- 不可避免地**通讯延迟**会导致车辆在协作时无法收到实时信息。
- 通讯可能会遭到**恶意干扰**

因此,在未来的研究中**可靠的通讯方式**和**高鲁棒性的协作系统**会是一个热点方向。

2.1.2 跨模态技术的应用

当前的大多数协作方式都是基于激光雷达的**3D点云信息**。然而,在实际应用中很多可用于感知的有价值的信息,例如声音信息,被忽略了。未来的发展可以向**多模态信息融合**的方向发展。

2.1.3 大规模数据集

在当前深度学习研究环境下,数据集已经成为继算法和算力之后,第三个限制技术发展的因素。在领域内出现**公认的,大规模的,高质量的数据**对自动驾驶领域的发展至关重要。

Learning Distilled Collaboration Graph for Multi-Agent Perception

3 论文概述

本文是一篇利用群体智能的协同感知领域的论文。文中提出了一种新的协作结构叫做 **Distilled collaboration graph (DiscoGraph)** 来实现各智体的**模型训练, 位置感知和协作**。

本文方法主要有两个创新点。

1. 使用了**教师-学生架构**, 即知识蒸馏技术。其中, 学生模型间使用中期协作策略, 可以**共享 Feature**; 而教师模型可以获得**所有个体的全部传感器信息**, 即拥有**全局信息**。通过学生-教师架构, 可以让部署在单个智体上具有共享性质的 **Distilled collaboration network (DiscoNet)** 接近教师模型的表现。
2. 提出了一种新的单位间权重度量手段即**矩阵边权 (matrix-value edge weight)**, 用来反映高分辨率下两个智体间传递的信息的重要程度。可以让一个智体适应性地突出需要关注的**空间信息**, 同时也可以使每个单位选择**合适的伙伴单位**获取有价值的信息。

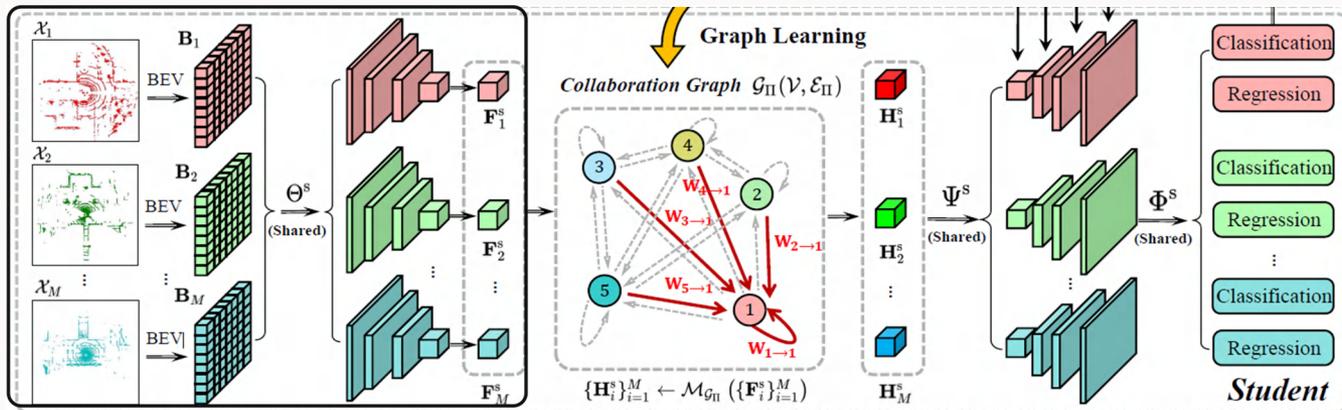


图 2: Student part of teacher-student training framework

3.1 学生模型 (中期协作)

本页介绍图2中黑框内步骤。学生模型首先将由自身传感器收集到的信息 \mathcal{X}_i 转化为 **BEV 图** $\mathbf{B}_i \in \{0, 1\}^{K \times K \times C}$, 然后通过**编码器**进行编码和压缩 $\mathbf{F}_i^s \leftarrow \Theta^s(\mathbf{B}_i)$ 得到**特征图** F_1^s , 其中 $\Theta^s(\cdot)$ 表示编码器。此时的特征图的形状信息为 $\mathbf{F}_i^s \in \mathbf{R}^{\bar{K} \times \bar{K} \times \bar{C}}$, 其中 $\bar{K} \times \bar{K}$ 为分辨率, \bar{C} 为通道数。

此时为了减少带宽占用, 作者团队在此处对特征图进行压缩。通过一个 1×1 的**自编码器 (AE)** 沿第二个维度进行降维。

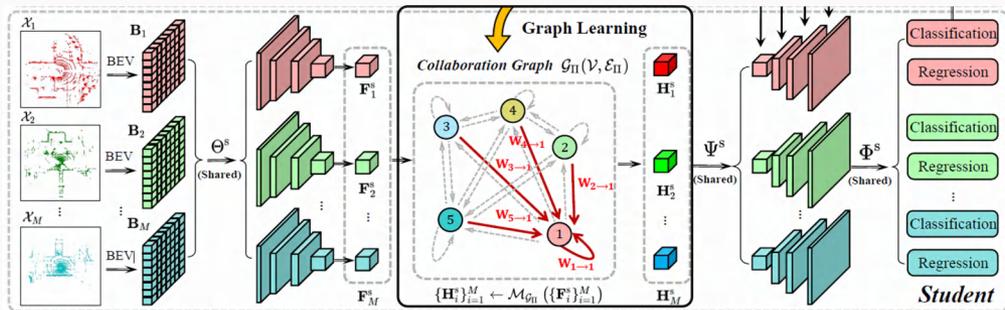


图 3: Student part of teacher-student training framework

本页介绍图3中黑框内步骤。在本页介绍的内容中,核心部分就是协作图 $\mathcal{G}_{\Pi}(\mathcal{V}, \mathcal{E}_{\Pi})$ 。其中 \mathcal{V} 表示点集,而 \mathcal{E}_{Π} 表示可训练的边集。令 $\mathcal{M}_{\mathcal{G}_{\Pi}}(\cdot)$ 为协作过程,则经协作图更新的特征图可被表示为 $\{\mathbf{H}_i^s\}_{i=1}^M \leftarrow \mathcal{M}_{\mathcal{G}_{\Pi}}(\{\mathbf{F}_i^s\}_{i=1}^M)$ 。而协作过程 $\mathcal{M}_{\mathcal{G}_{\Pi}}(\cdot)$ 由三个过程构成,即神经消息传递 (S1),神经消息处理 (S2) 和神经消息融合 (S3)。

1. **神经消息传递 (S1)**: 在该阶段各单位都向其他的单位传输自己的特征图。
2. **神经消息处理 (S2)**: 在该阶段各单位需要根据其他单位的特征图计算两个单位间的边权。从 j 单位收到的经转换的特征图可被表示为 $\mathbf{F}_{j \rightarrow i}^s = \Gamma_{j \rightarrow i}(\mathbf{F}_j^s) \in \mathbf{R}^{\bar{K} \times \bar{K} \times \bar{C}}$, 其中转换过程 $\Gamma_{j \rightarrow i}(\cdot)$ 中包含了 i 与 j 的相对位置。此时利用 $\mathbf{F}_{j \rightarrow i}^s$ 和 \mathbf{F}_i^s 计算两个单位间的边权。需要注意的是,此处的边权并不是传统意义上的两个点间的距离或代价,而是用二维权重矩阵表示的两单位间信息的重要性。这个矩阵可被表示为 $\mathbf{W}_{j \rightarrow i} = \Pi(\mathbf{F}_{j \rightarrow i}^s, \mathbf{F}_i^s)$, 其中 Π 将两个矩阵沿第二维进行拼接,然后使用四个 1×1 卷积层将第二维抹除,此时两个三维的特征图变成了一个二维的权重矩阵。这个步骤也是我优化的部分。
3. **神经消息融合 (S3)**: $\mathbf{H}_i^s = \sum_{j=1}^M \mathbf{W}_{j \rightarrow i} \odot \mathbf{F}_{j \rightarrow i}^s$ 将从各单位获得的权重矩阵 $\mathbf{W}_{j \rightarrow i}$ 与 $\mathbf{F}_{j \rightarrow i}^s$ 求 Hadamard

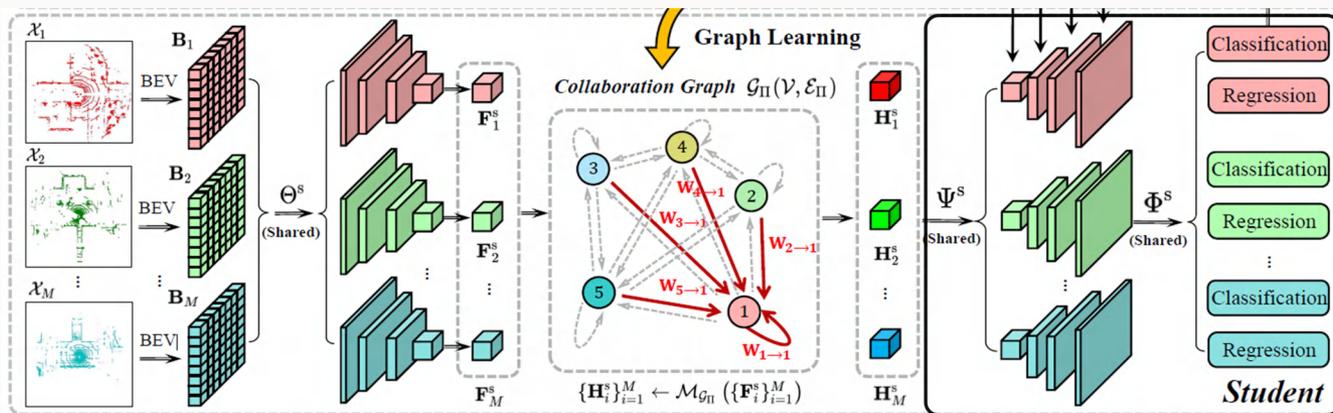


图 4: Student part of teacher-student training framework

乘积再求和, 获得经其他单位更新后的特征图。

本页介绍图4中黑框内步骤。

在本页中介绍学生模型中的收尾阶段即解码器和分类头。

每一个单位使用解码器 $\Psi^s(\cdot)$ 对更新后的特征图解码 $\mathbf{M}_i^s \leftarrow \Psi^s(\mathbf{H}_i^s)$ 。解码器使用四层实现上采样, 每一层首先将原特征图与经更新后的特征图拼接, 然后使用 1×1 卷积对通道数减半。此时, \mathbf{M}_i^s 与 \mathbf{H}_i^s 及 \mathbf{F}_i^s 均已对齐。

最后, 使用分类头对解码后的特征图进行分类 $\hat{\mathbf{Y}}_i^s \leftarrow \Phi^s(\mathbf{M}_i^s)$ 。分类头使用两个卷积层分支来对前景-背景类别进行分类, 并进行边界框回归。

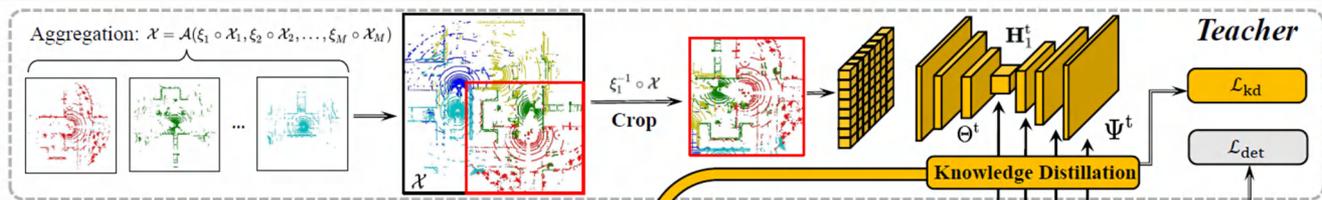


图 5: Teacher part of teacher-student training framework

3.2 教师模型 (早期协作)

在训练阶段, 有一个**早期协作模型**作为教师模型来指导学生模型。令 $\mathcal{X} = \mathcal{A}(\xi_1 \circ \mathcal{X}_1, \xi_2 \circ \mathcal{X}_2, \dots, \xi_M \circ \mathcal{X}_M)$ 为**全局视角**的 3D 点云。其中 $\mathcal{A}(\cdot, \dots, \cdot)$ 是将多个 3D 点云的融合操作, ξ_i 和 \mathcal{X}_i 分别表示单位 i 的位置信息和该单位收集到的 3D 点云。此时, $\xi_i^{-1} \circ \mathcal{X}$ 和 \mathcal{X}_i 表示同一个区域的 3D 点云, 且是对齐的。

将 3D 点云 $\xi_i^{-1} \circ \mathcal{X}$ 转换为 BEV 图, 然后使用编码器提取出特征图 $\mathbf{H}_i^t \in \mathbf{R}^{\bar{K} \times \bar{K} \times \bar{C}}$ 与学生模型不同的是, 由于教师模型使用了拥有**全局视角**的早期协作策列, 因此并不需要根据其他单位传来的特征图对自身的特征图进行更新。

之后和学生模型一样使用解码器和分类头对前景-背景类别进行分类, 并进行边界框回归 $\hat{\mathbf{Y}}_i^t \leftarrow \Phi^t(\Psi^t(\mathbf{H}_i^t))$ 。

3.3 知识蒸馏

3.3.1 概述

在本文中, 使用知识蒸馏的主要目的是缩小模型体积从而保证带宽与性能平衡。这是知识蒸馏技术的一个典型应用, 在部分应用场景下, 受到存储和带宽的限制只能使用小模型。但是模型体积的缩小会带来致命的缺点: 泛化能力弱。而知识蒸馏技术就可以缓解这一问题。从图6可以看出, 传统的训练过程是对 ground truth 求极大似然, 而知识蒸馏的训练过程是对大模型的 class probabilities 求最大似然。这样, 带宽较小的学生模型在训练的过程中就可以学习到教师模型的泛化能力。

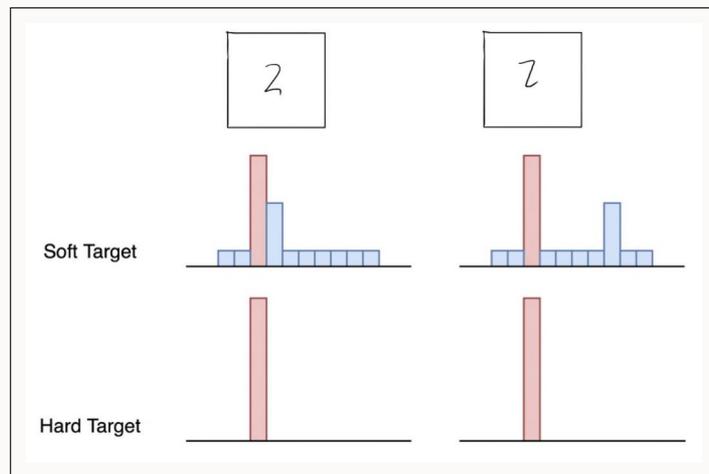


图 6: Comparison between KD's training process and traditional training process

3.3.2 本文方法

设教师模型已经训练完毕。我们使用 ground truth 和 knowledge distillation loss 加权共同计算机损失值来优化学生模型。即令：

$$\mathcal{L}^s = \sum_{i=1}^M \left(\mathcal{L}_{\text{det}}(\mathbf{Y}_i^s, \hat{\mathbf{Y}}_i^s) + \lambda_{\text{kd}} \mathcal{L}_{\text{kd}}(\mathbf{H}_i^s, \mathbf{H}_i^t) + \lambda_{\text{kd}} \mathcal{L}_{\text{kd}}(\mathbf{M}_i^s, \mathbf{M}_i^t) \right) \quad (1)$$

最小。

对于公式1中第一项, 包含前景-背景分类损失和边界框损失, 该损失值是学生模型的分头输出与 ground truth 计算得到的。公式1中第二项 $\lambda_{\text{kd}} \mathcal{L}_{\text{kd}}(\mathbf{H}_i^s, \mathbf{H}_i^t)$ 是学生模型经其他单位的特征图更新后的特征图与教师模型的特征图计算得到的损失值。公式1中的第三项 $\lambda_{\text{kd}} \mathcal{L}_{\text{kd}}(\mathbf{M}_i^s, \mathbf{M}_i^t)$ 是学生模型的解码器输出与教师模型的解码器输出计算得到的损失值。公式1中 $\mathcal{L}_{\text{kd}}(\mathbf{H}_i^s, \mathbf{H}_i^t)$ 被表示为

$$\mathcal{L}_{\text{kd}}(\mathbf{H}_i^s, \mathbf{H}_i^t) = \sum_{n=1}^{\bar{K} \times \bar{K}} D_{\text{KL}} \left(\sigma \left((\mathbf{H}_i^s)_n \right) \parallel \sigma \left((\mathbf{H}_i^t)_n \right) \right) \quad (2)$$

其中 D_{KL} 表示 KL 散度。 $\sigma(\cdot)$ 表示在将特征向量在通道维度做 softmax 操作。

3.4 实验设置

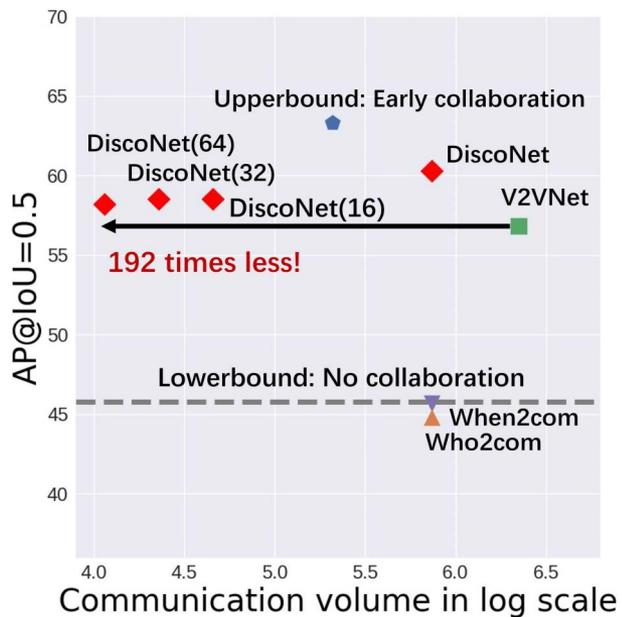
作者在本文主要设置了两组实验。第一组实验是将本文提出的方法与传统的方法进行对比, 对比的指标包括准确率和带宽。第二组实验是消融实验, 证明本文提出的协作策略和知识蒸馏技术的有效性。

3.4.1 定量评估

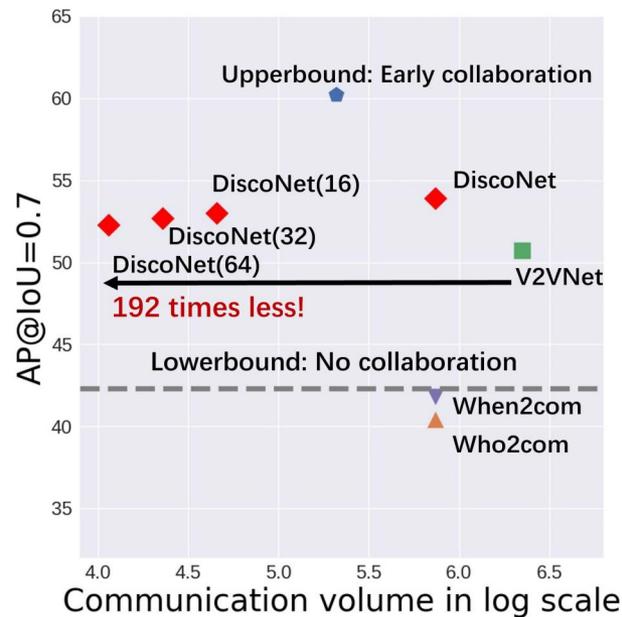
Method	Collaboration Approach			Average Precision (AP)	
	Early	Intermediate	Late	IoU=0.5	IoU=0.7
Upper-bound	✓	✗	✗	63.3	60.2
	✓	✗	✓	59.7	55.8
When2com* [19]	✗	✓	✗	45.7	41.7
When2com [19]	✗	✓	✗	45.7	41.8
Who2com* [20]	✗	✓	✗	44.3	40.3
Who2com [20]	✗	✓	✗	44.8	40.4
V2VNet [34]	✗	✓	✗	56.8	50.7
DiscoNet	✗	✓	✗	60.3	53.9
DiscoNet(16)	✗	✓	✗	58.5	53.0
Lower-bound	✗	✗	✓	57.6	54.2
	✗	✗	✗	45.8	42.3

图 7: 各方法在目标检测问题上的准确率对比。* 表示有位置感知。本文方法标粗体。其中 **DiscoNet(16)** 表示在自编码器部分经过 16 次压缩。

其中, lower-bound 是单智体, 而 upper-bound 是具有全局视角的教师模型。两者都使用了非极大值抑制 (NMS)。i) 第一行相较于最后一行准确率显著提升, 证明了早期协作的有效性。ii) 晚期协作提升了 lower-bound 的准确率却降低了 upper-bound 的准确率可能的原因是 NMS 可以移除多余的噪声框但是也可能移除了有用的框。iii) 在所有方法中, 本文方法表现最好。



(a) Scatterplot in AP@IoU 0.5



(b) Scatterplot in AP@IoU 0.7

图 8: 各方法的准确率-带宽图。x 轴为带宽。y 轴为准确率

从图8看出 i) 经过压缩的 Disconet 的通信量显著低于为压缩的 Disconet。ii) 即使经过 64 次压缩的 Disconet 的准确率依然要高于 V2VNet。iii) 压缩并没有导致准确率显著下降。

3.4.2 定性评估

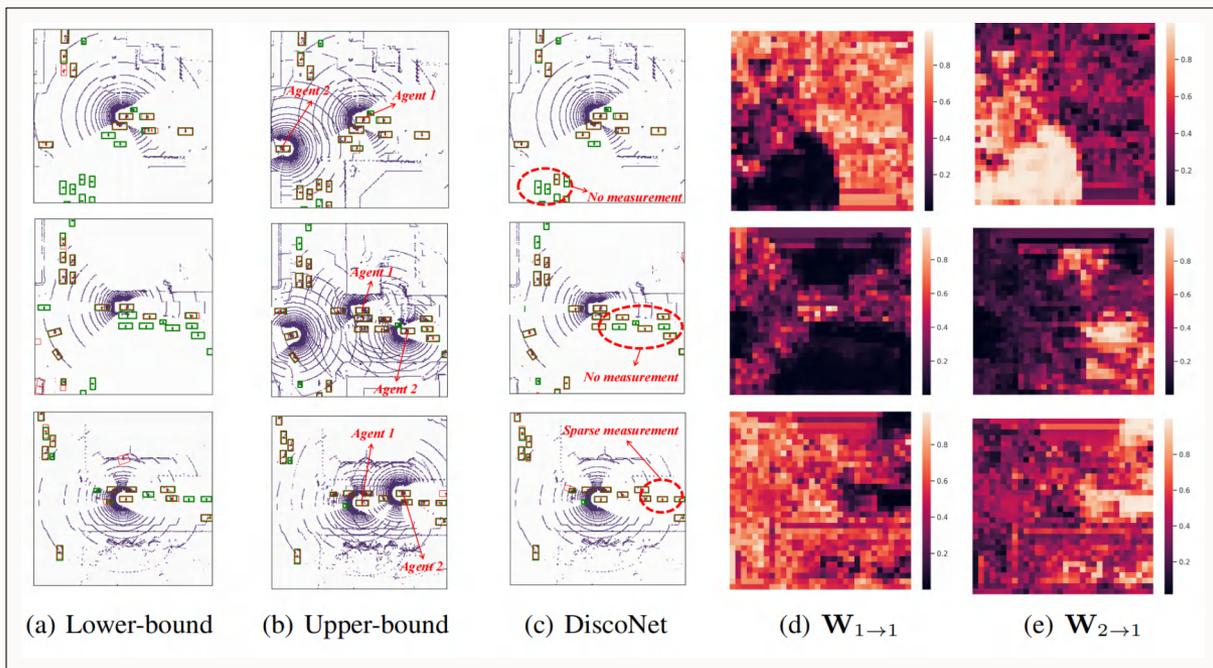


图 9: 图中绿框和红框分别表示真实值和预测值。图 d 是单位一的矩阵边权。图 e 是单位二传输给单位一的矩阵边权, 补充了单位一没有观察到的信息。

观察图9可以看出相较于 (a), (c) 表示的 DiscoNet 在被遮挡和远距离的物体的检测中表现更好。这一现象可以用图 (d) 和图 (e) 来解释。观察图 (d) 的第一张图, 该图的左下角有一片黑色区域, 该区域的置信度较低。而从图 (e) 的第一张图可以看出, 单位一向单位二请求的信息中在该区域的注意力显著高于其他区域。

3.5 消融实验

Collaboration Strategy	Scalar Weight				Matrix Weight		
	No	Sum	Average	Weighted Average	Max	Cat	DiscoGraph
w/o KD (AP@IoU 0.5/0.7)	45.8/42.3	55.7/50.9	55.7/50.4	56.1/51.4	56.7/51.4	55.0/50.2	57.2/52.3
w/ KD (AP@IoU 0.5/0.7)	46.5/42.9	54.4/46.3	56.4/51.1	56.7/50.9	56.7/51.8	57.5/52.6	60.3/53.9
Gain	0.7/0.6	-1.3/-4.6	0.7/0.7	0.6/-0.5	0.0/0.4	2.5/2.4	3.1/1.6
Gain (%)	1.5/1.4	-2.3/-9.0	1.3/1.4	1.1/-1.0	0.0/0.7	3.3/4.6	5.4/3.1

图 10: 中期协作策略比较

表10分为三个部分。最左侧一列表示是否使用知识蒸馏技术。中间一列表示在协作中使用标量权重。最右侧一列表示使用矩阵权重。

1. 对比蓝框中的内容, 知识蒸馏技术有效提高了感知的准确率。
2. 通过对比第一列即红框, 可以看出本文提出的方法 DiscoGraph 在策略上是最优的。
3. 对于没有学习能力的 Max 列来说, 是否使用知识蒸馏对其准确率并没有影响。排除了知识蒸馏技术可能在其他方面影响实验结果的可能性。

4 代码实践

4.1 实验环境

OS	Win10
GPU	3090(资深矿工)
CPU	i5-13600kf
RAM	64GB

4.2 实验步骤

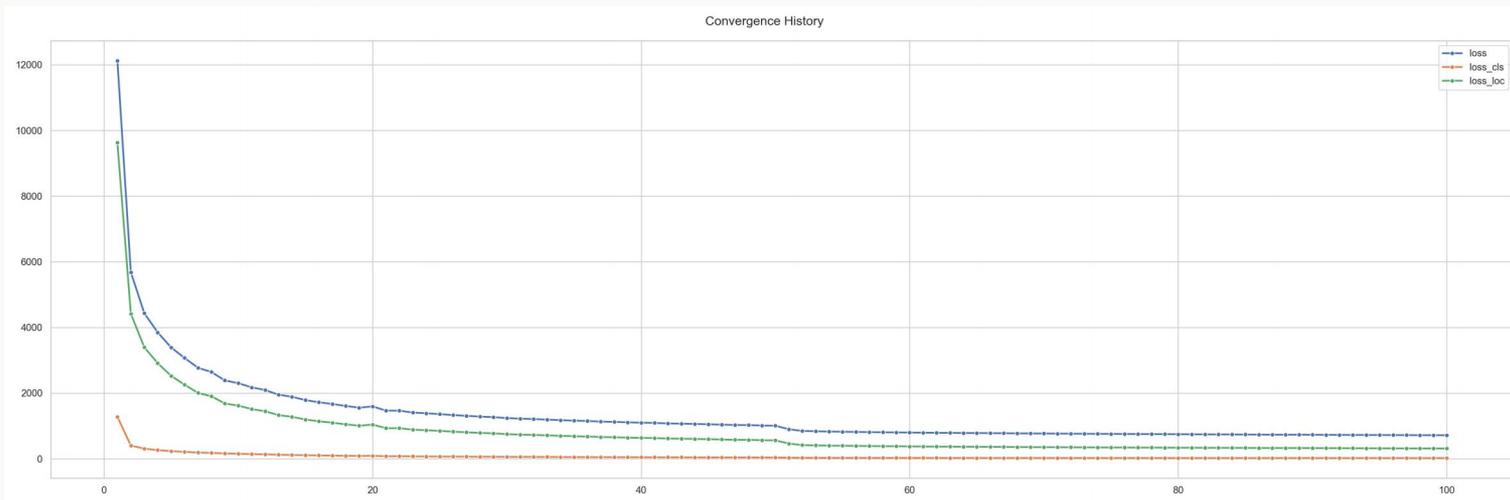
1. 使用 `git clone` 从 github 上下载代码
2. 从谷歌云盘下载数据集, 但是下载了两天都因为网络问题失败, 于是去淘宝找了代下通过邮箱发给我
3. 使用 `conda` 搭建实验环境, 利用 `requirements.txt` 文件一键安装环境, 但是出现了第三方库版本不兼容的情况, 于是根据 [Stack Overflow](#) 的帖子和 [Github](#) 上的 `issue` 配置版本。
4. 由于第三方库的版本的更新, 需要对文件中一些代码进行更改。
5. 配置形参, 训练检测模型

4.3 模型训练

形参:

```
1 --data D:\Project\bputSC\data\V2X-Sim-det\train
2 --com disco
3 --save_freq 10
4 --batch_size 4
5 --log
6 --kd_flag 1
7 --resume_teacher D:\Project\bputSC\checkpoints\det\upperbound\with_rsu\epoch_100.pth
8 --auto_resume_path logs
9 --logpath logs
10 --nepoch 100
11 --rsu 1
```

Loss 曲线 (100 个 epoch):

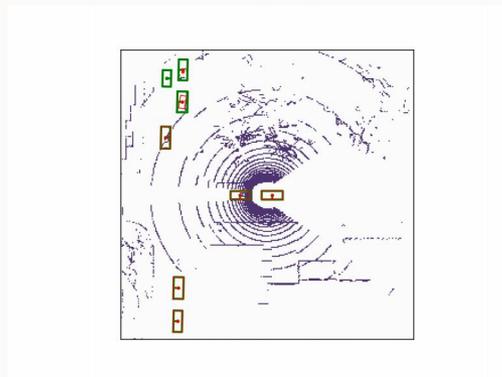


4.4 测试结果

表 1: Average Precision

	IoU=0.5	IoU=0.7
Agent 1	0.65	0.61
Agent 2	0.75	0.70
Agent 3	0.76	0.67
Agent 4	0.73	0.64
Agent 5	0.77	0.71
平均准确率	0.72	0.66

4.5 样例抽取



(a) 鸟瞰图

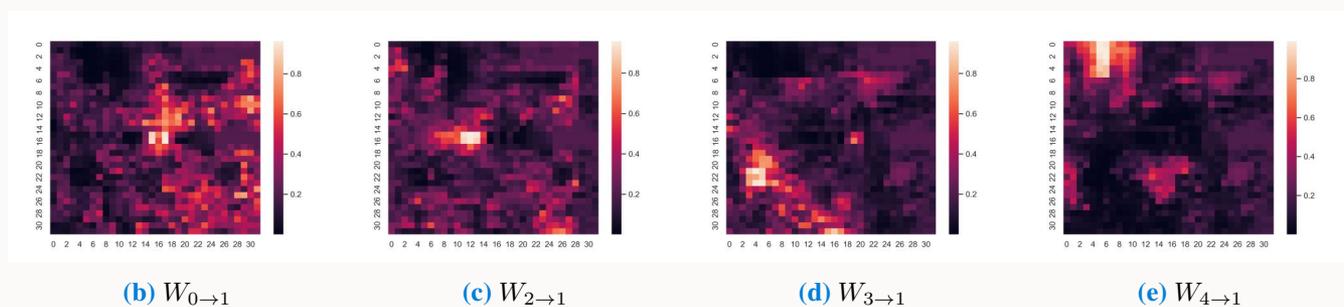


图 11: 通过子图 (a) 发现在单位 1 的左上角有两个车辆没有被正确识别。观察 (b),(c),(d),(e), 只有 (e) 的左上角为白色, 其他的子图都是黑色。以图 (b) $W_{0 \rightarrow 1}$ 为例, 该图左上角为黑色, 表示 1 在对 0 发送的特征图的注意力并没有关注到左上角。这也是我的优化思路之一。

5 算法优化

5.1 思路一：特征融合

在上一段我们提到, 样例11左上角的两个车辆没有被正确识别是 DiscoNet 在特征融合阶段的缺陷导致的。通过观察源码我发现, 原论文的特征融合是将自身的特征与其他智体的特征在第二维度拼接 ($[1, 512, 32, 32]$), 然后使用卷积核大小为 1×1 的卷积神经网络进行特征提取 ($[1, 1, 32, 32]$)。

我认为这个过程是有局限的, 卷积神经网络虽然将 512 层提取至 1 层在第二个维度上实现了特征融合, 但是并没有在第三第四维度上进行特征融合。因此, 在使用卷积神经网络进行特征提取前, 我希望找到一种手段在两种特征第三第四维度上也进行特征融合。于是我在特征融合操作前使用了自注意力机制。

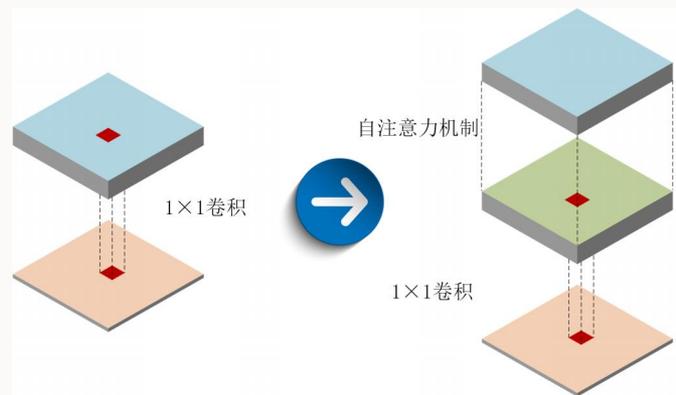


图 12: 在特征融合操作前加入了自注意力机制

5.1.1 遇到的问题

首先是矩阵对齐的问题,在不改变输入矩阵和输出矩阵的形状下实现注意力机制浪费很长的时间。

其次是硬件问题,加入了注意力机制后程序对内存的占用陡然上升,只好买两根新的内存条。装上新的内存条后,我发现显存占用只有 14GB,秉持着不浪费硬件资源的原则我增加了 BATCHSIZE 值,但是此时内存又占满了,只好再把 BATCHSIZE 改回去。

即使是这样,依然时常出现蓝屏和重启的状况,为了保证训练的顺利完成此后的实验我都只进行了十轮训练。



5.1.2 Loss 曲线对比

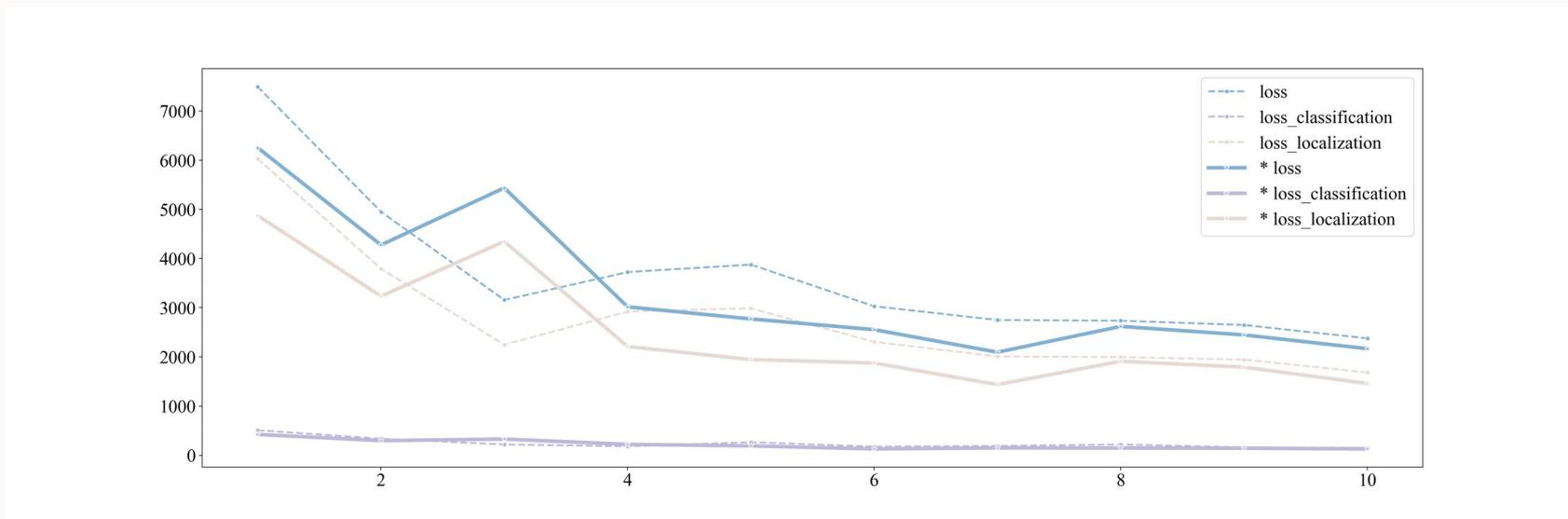


图 13: Loss 曲线对比。带 * 号为改进后的方法。loss, loss_classification, loss_localization 分别代表总损失, 分类损失, 定位损失。

观察图 13 可以看出, 改进后的方法对分类损失没有显著影响, 但是对定位产生了较大的影响。loss_localization 的值除了在第二个 epoch 外都低于 loss_localization 的值。

这初步证实了我们的设想, 使用注意力机制可以**提高特征融合步骤中智体对其他智体传来的特征图关键位置的注意力**, 从而提高对环境中其他车辆的定位准确率。

5.1.3 思路一总结

在思路一中, 我们根据 4.5 章发现的 DiscoNet 对其他智体传来的特征图的关键位置注意力不集中的问题提出了猜想即认为这种不集中可能是在特征融合的过程中仅仅在第二维度进行了特征融合, 而忽视了第三和第四维度的信息。当时我的第一想法是放弃 1×1 卷积而使用 3×3 或 5×5 卷积, 这样就可以在后三个维度上进行信息融合。但是使用 3×3 或 5×5 卷积也会有缺点和局限。

- 缺点是会改变第三和第四维度的形状, 导致后续运算需要大面积的代码修改, 还可能遇到未知的问题。
- 局限是在于卷积是一种局部视野特征提取的方法, 这种方法或许在图像领域有很高的适应性, 因为自然的画面一般是连贯的, 但是在自动驾驶领域我们更需要关注的是远距离和全局的信息, 因此我认为使用自注意力机制会更加合适。

综上, 我在特征融合步骤中加入了**自注意力机制模块**。经实验证明, 该方法可以有效降低定位损失值, 证明了该猜想的合理性和该方法的有效性。然而, 本方法也有不足之处。首先, 最直接的不足就是会提高训练的时间成本, 相比于原方法, 改进后的方法每一个 epoch 要多运算 1 分钟左右。其次, 本文方法会显著提高内存占用率, 这一点的原因目前还没有查明, 但是可能与并行矩阵运算有关。

References

- [1] 罗贵阳. 车路协同环境下通信资源调度方法研究 [D]. 北京邮电大学,2020.DOI:10.26969/d.cnki.gbydu.2020.000025.
- [2] 栗颂乔. 基于车云融合的车辆编队协同机制研究 [D]. 北京邮电大学,2021.DOI:10.26969/d.cnki.gbydu.2021.000407.
- [3] 张皓涵. 基于群体智能的车联网资源优化研究与实现 [D]. 北京邮电大学,2020.DOI:10.26969/d.cnki.gbydu.2020.001313.
- [4] 胡煜. 面向车路协同感知的车辆重识别与轨迹预测方法研究 [D]. 江苏大学,2022.DOI:10.27170/d.cnki.gjsuu.2022.001787.
- [5] Arnold E, Dianati M, de Temple R, et al. Cooperative perception for 3D object detection in driving scenarios using infrastructure sensors[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 23(3): 1852-1864.
- [6] Li Y, Ren S, Wu P, et al. Learning distilled collaboration graph for multi-agent perception[J]. Advances in Neural Information Processing Systems, 2021, 34: 29541-29552.
- [7] 葛元, 张为峰, 韩延涛等. 基于 C-V2X 的车路群体协同混合组网 [J]. 无线电工程,2022,52(01):11-18.
- [8] 邢亚男. 车路协同感知融合研究 [D]. 吉林大学,2022.DOI:10.27162/d.cnki.gjlin.2022.006243.

谢谢观看