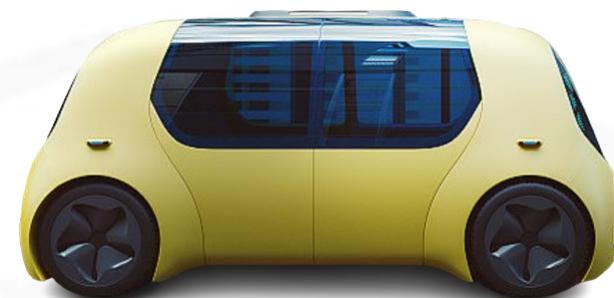




目标跟踪& 轨迹预测 专题



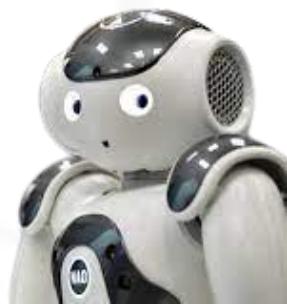
什么才是理想要求的智能无人系统？



能听，会看

能动，会说

有思考，会推理





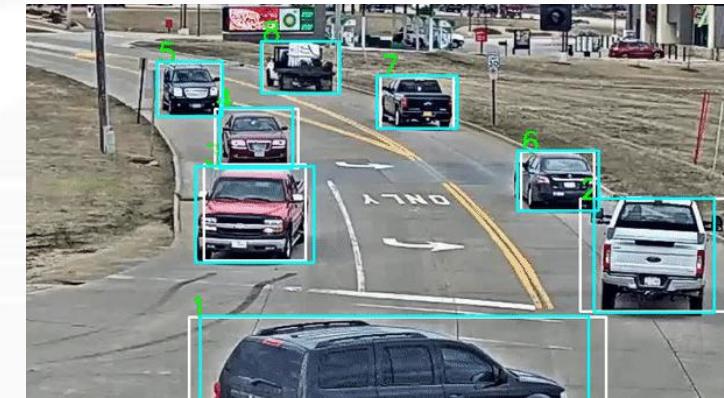
目标跟踪 Object Tracking



Sport Scenarios



Surveillance Scenarios



Driving Scenarios



目录



1. 背景概述



2. 经典方法



3. 研究方向



4. 研究数据



5. 评价指标



6. 应用领域



7. 未来展望

1. 背景概述

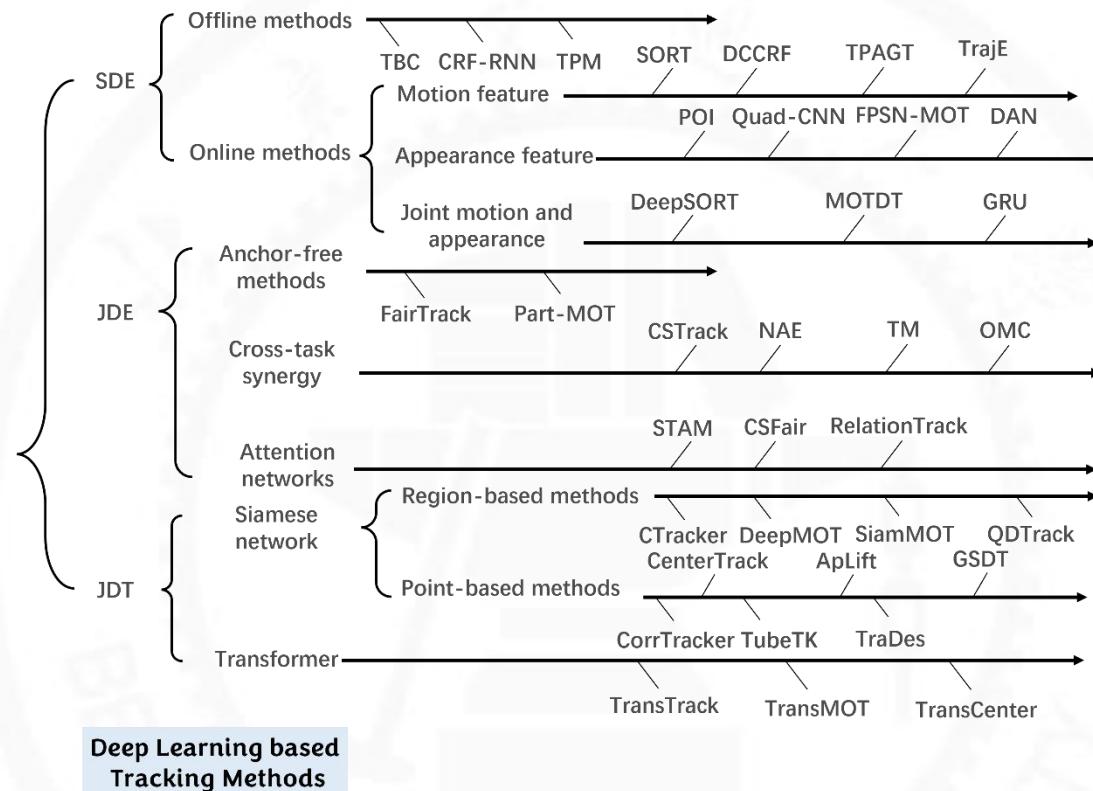
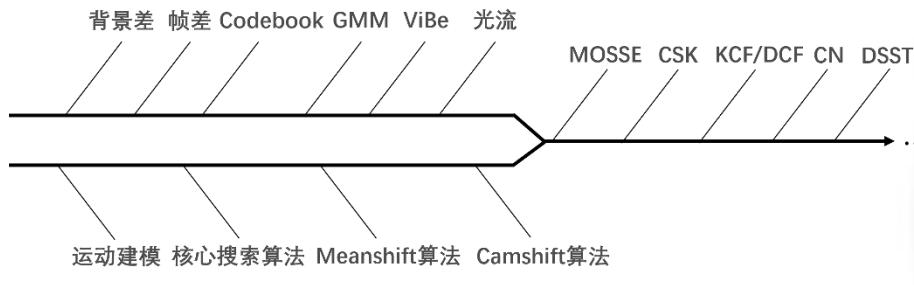


北京交通大学

目标跟踪

- ▲ 基本定义：目标跟踪是指在视频或图像序列中，通过对目标进行连续观察和分析，从而实现对目标在时间和空间上的持续追踪和定位的过程；
- ▲ 简单分类：基于传统滤波方法和基于深度学习的方法；
- ▲ 研究意义：实现对目标的连续感知与定位，为视觉应用提供实时的目标分析、交互体验和智能决策支持；
- ▲ 发展历程：

Object Tracking Milestones

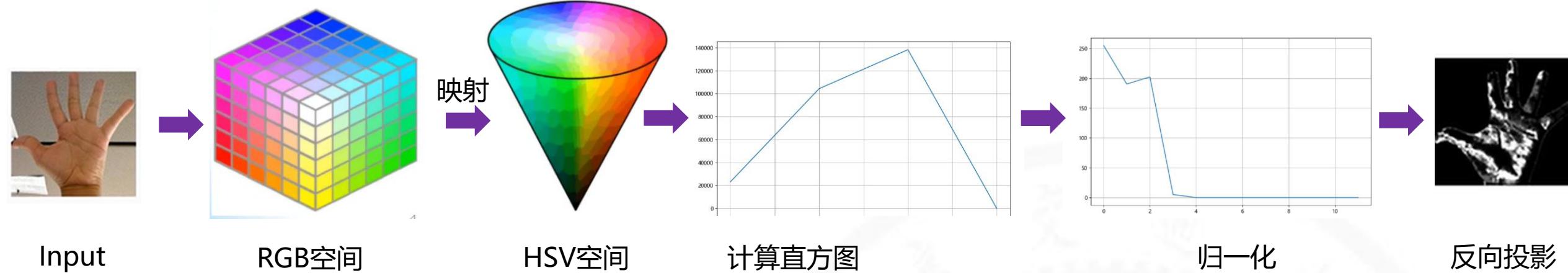


2. 经典方法——传统方法

(1) MeanShift



北京交通大学



Input

RGB空间

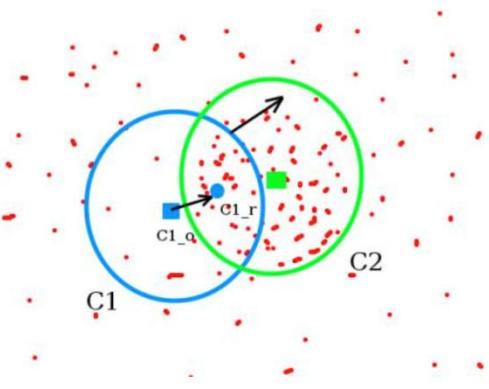
HSV空间

计算直方图

归一化

反向投影

- 根据初始目标box的大小的位置确定meanshift搜索窗口的大小和位置
- 计算搜索窗口内的质心，或叫加权平均位置
- 求得的质心作为新的搜索窗口的中心



MeanShift其实关键:

- mean**，质心（也可以叫加权平均值）
- shift**，即向质心方向移动



2. 经典方法——传统方法

(2) Camshift: Continuously Adaptive Mean-Shift

mean-shift算法的窗口是固定大小的，而我们需要让**窗口大小和目标的大小**以及**旋转相适应**，因此提出了Camshift



- 第一帧的目标跟踪box位置和大小都是已知的
- 根据meanshift迭代计算出第二帧的位置信息
- 由于不变矩的存在，这两者可以计算出相同的二阶矩
- 然后代入第二帧的位置信息, 逆推出第二帧的大小信息

$$M_{20} = \sum_x \sum_y x^2 I(x, y)$$



存在着目标被其他**物体遮盖**的现象，导致目标框无法准确的进行跟踪

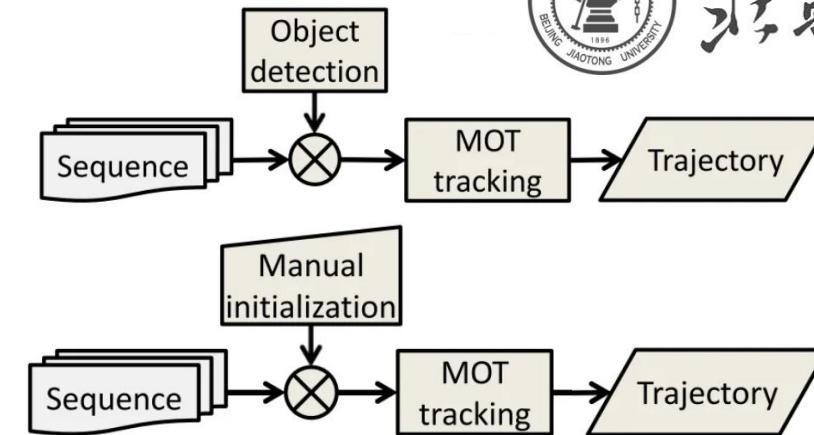
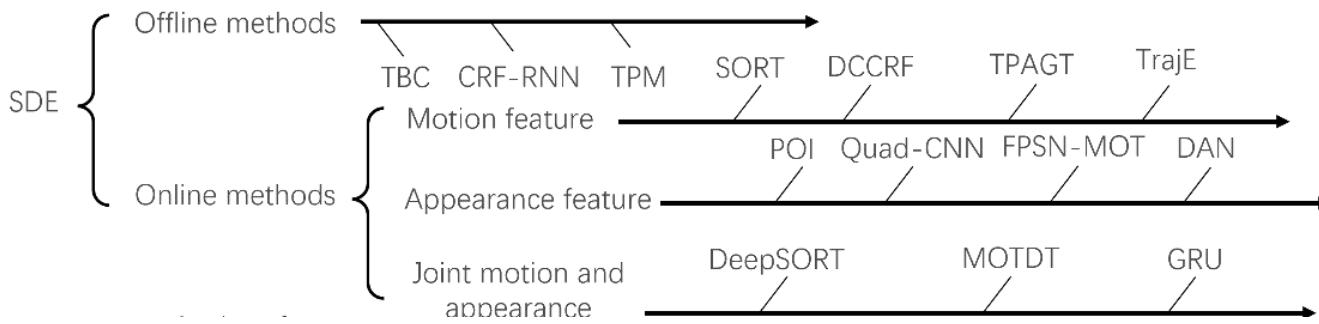


北京交通大学

2. 经典方法——深度学习方法

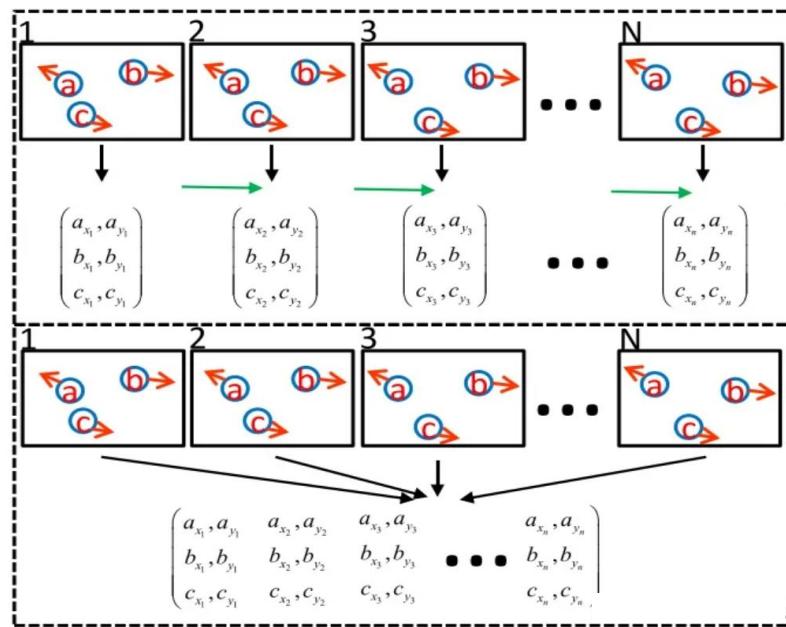


北京交通大学

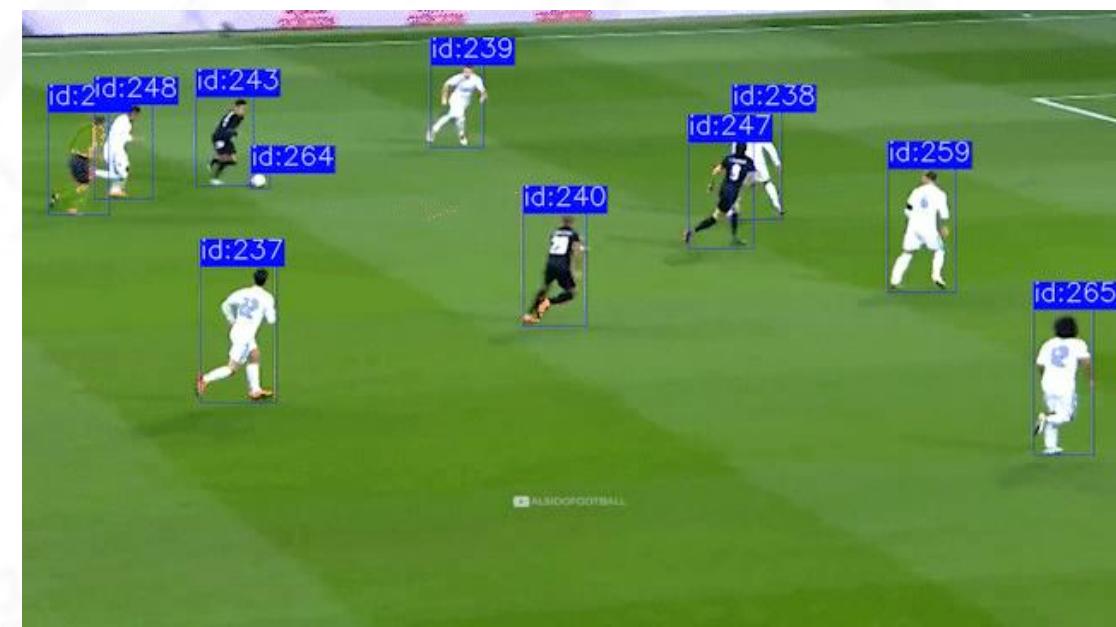


分离检测与特征提取的方法 (SDE范式)

在线方法



离线方法

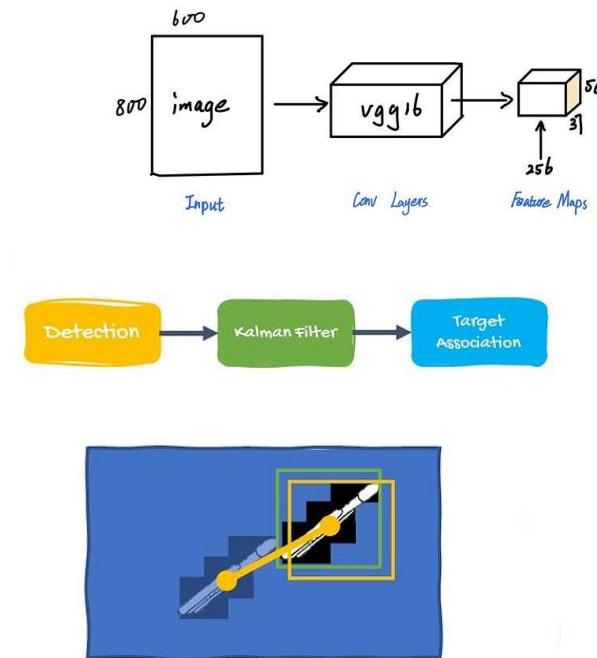


2. 经典方法——深度学习方法

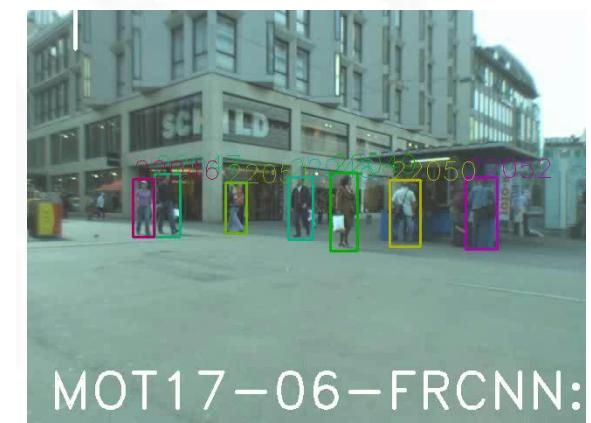
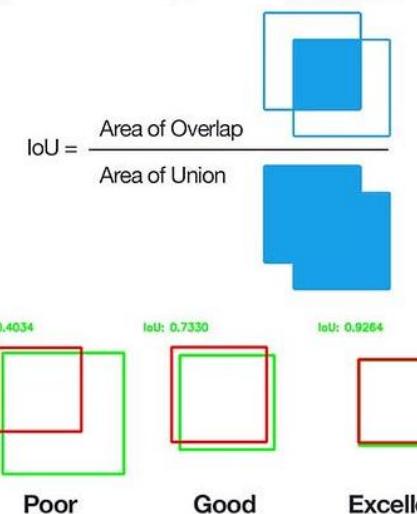
(1) SORT: SIMPLE ONLINE AND REALTIME TRACKING

- **Detection**
- **Estimation**
- **Association**
- **Track Identity creation and destruction**

Better Detection = Better Tracking



- 使用速度恒定模型将检测从当前帧传播到下一帧
- 在将检测分配给现有目标时，每个目标的边界框几何形状是通过预测其在最新帧中的新位置来估计的
- 计算每个检测与现有目标的所有预测边界框之间的交并 (IOU)
- 使用匈牙利算法最优地解决分配问题



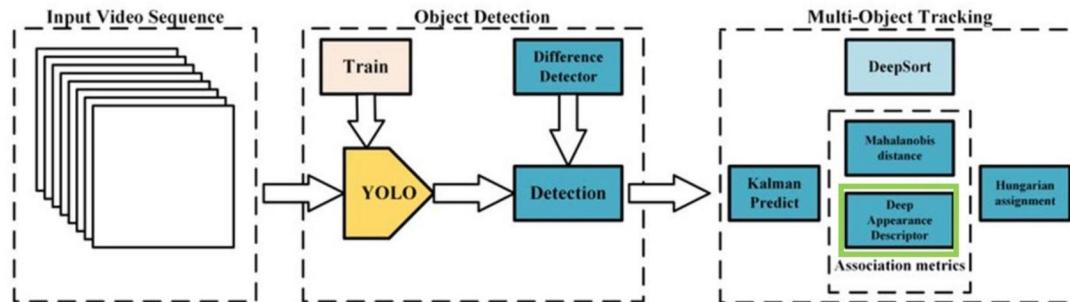
2. 经典方法——深度学习方法



北京交通大学

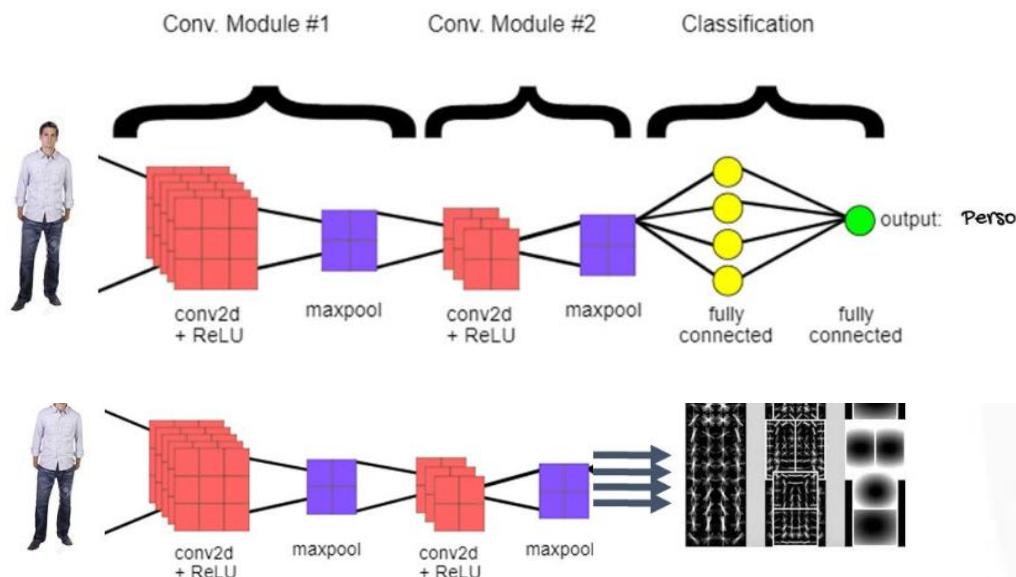
(2) DeepSORT

由于SORT返回较多数量的**identity switches**, 并且在通过**遮挡**和**不同视点**等进行跟踪方面存在缺陷



- 卡尔曼滤波器预测轨迹 Tracks
- 使用匈牙利算法将预测得到的轨迹 Tracks 和当前帧中的 detections 进行匹配 (级联匹配和 IOU 匹配)
- 卡尔曼滤波更新

外观特征向量

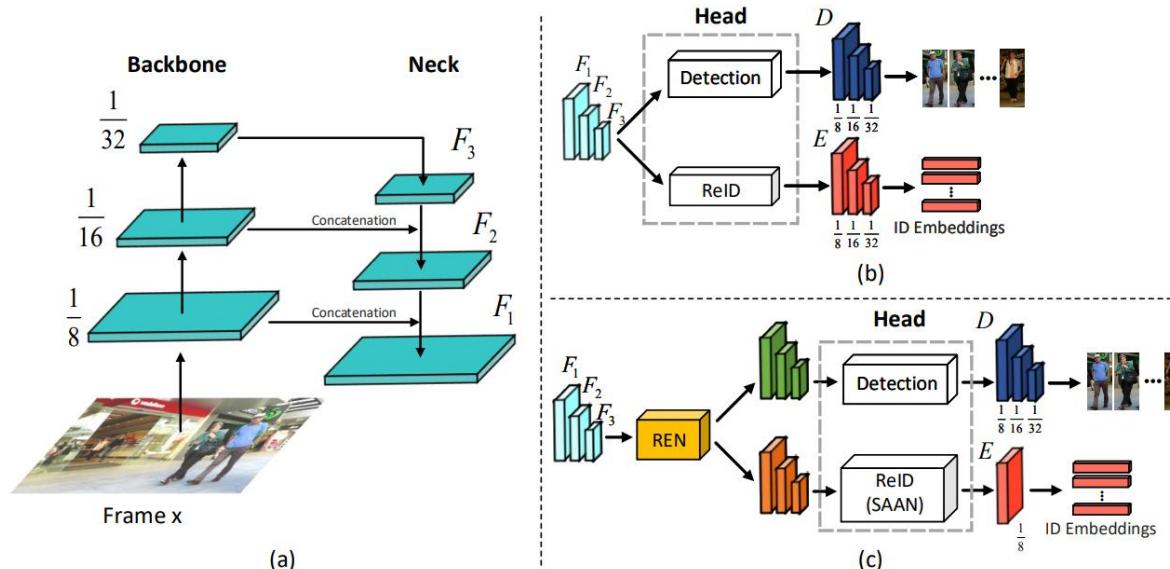


- 计算相似度矩阵的方法使用到了外观模型 (ReID) 和运动模型
- 匹配过程是一个循环, 没有丢失过的轨迹优先匹配, 丢失较为久远的就**靠后匹配**
- 假如一个轨迹要去匹配两个表观特征非常相似的 Detection, 分别让两个 Detection 计算与这个轨迹的马氏距离, 并使用一个**阈值 threshold** 进行限制

2. 经典方法——深度学习方法

(3) CStrack

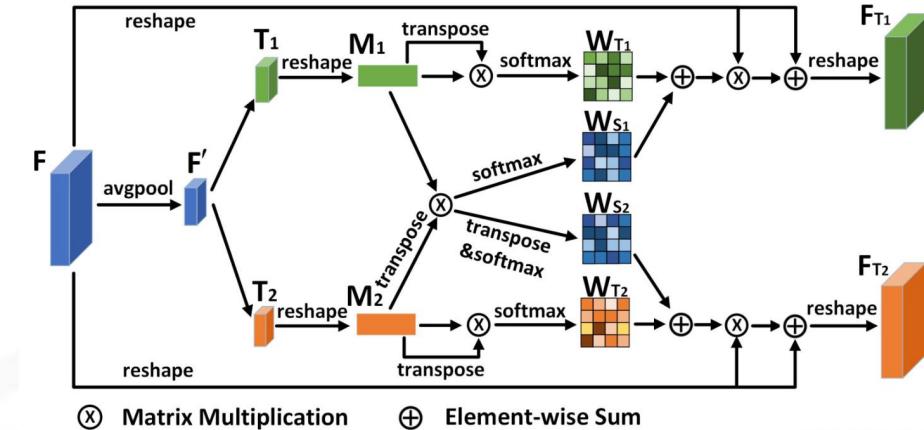
Two-stage的检测模型不小且 ReID 模型需要在每个检测框上进行推理，计算量非常之大，所以 JDE 的诞生是一种必然



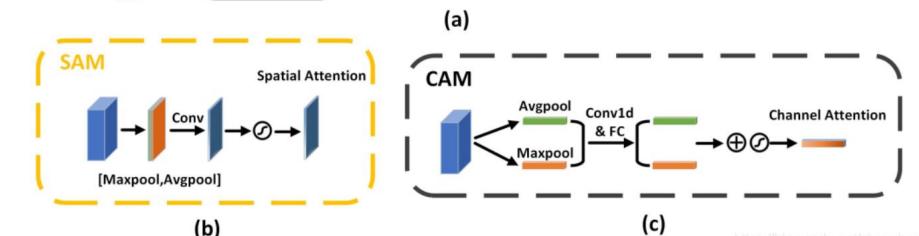
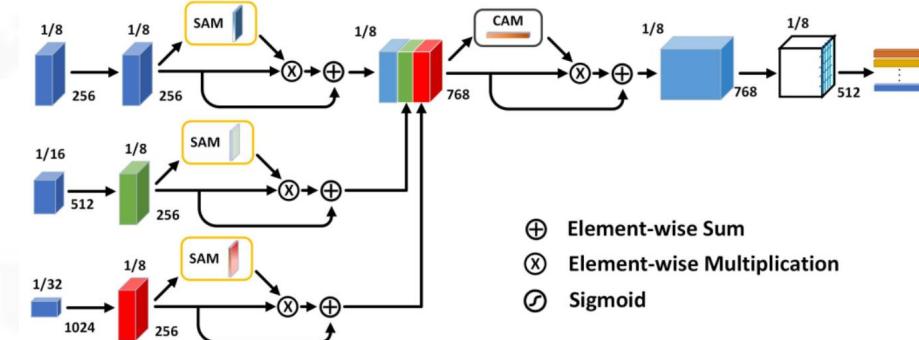
- 将 **Detection** 和 **ReID** 解耦为两个分支，分别学习
- 两个任务的特征通过自注意力方式获得**自注意力权重图**和**交叉相关性权重图**
- 自注意力图是促进各自任务的学习，交叉相关图是为了提高两个任务的协同学习
- 为了解决尺度问题，设计了**尺度感知注意力网络 (SAAN)** 用于 ReID 特征的进一步优化
- 不同尺度外观特征融合输出



交叉相关网络 (Cross-correlation Network, CCN)



Scale-aware Attention Network, SAAN

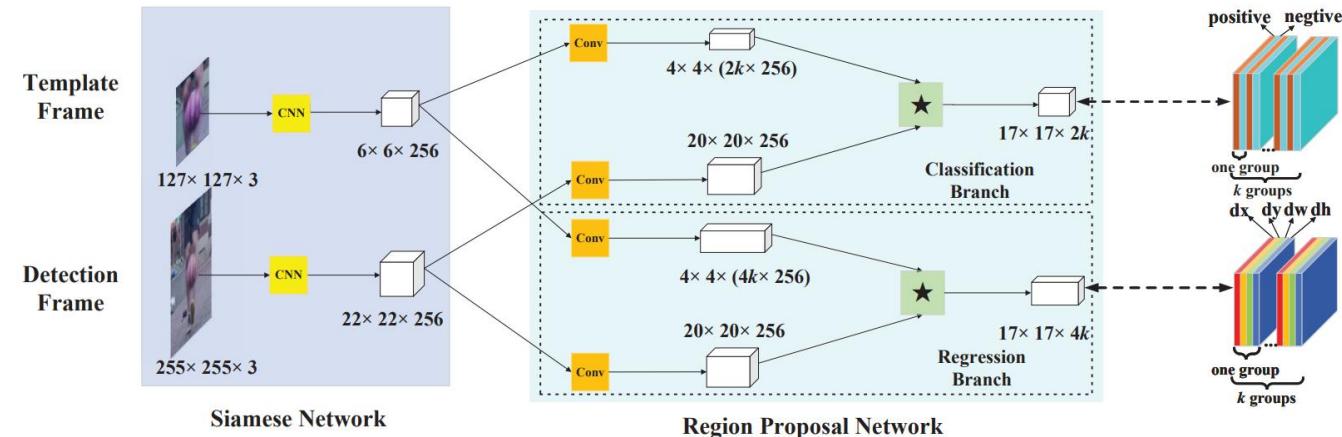
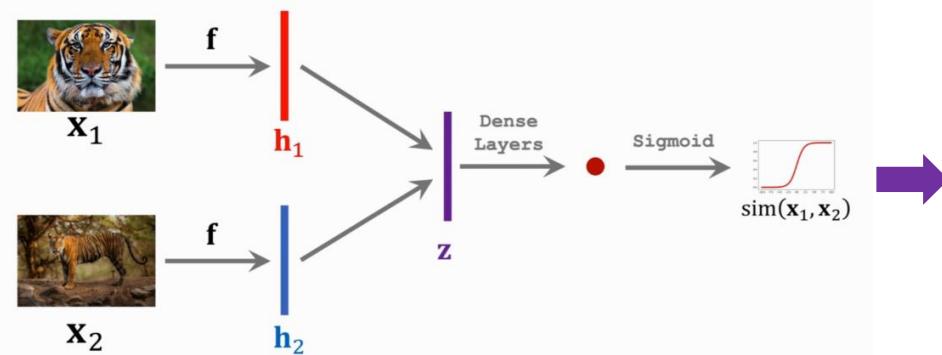


2. 经典方法——深度学习方法

(4) Siamese based: SiameseRPN



北京交通大学



将输入数据同时输入到两个完全相同的神经网络中

将RPN网络和经典的Siamese进行结合

$$\min_W \frac{1}{n} \sum_{i=1}^n \mathcal{L}(\zeta(\varphi(x_i; W); \varphi(z_i; W)), \ell_i)$$

1. 第一帧中截取尺寸为(127, 127, 3)的Template Image, 下一帧中截取Search Image, 将Template和Search分别送入alexnet提取特征
2. Teamplate和Search通过Alexnet提取特征
3. 特征送入RPN网络的class分支, 输出尺寸为(1, 10, 17, 17)的预测结果, 其中10表示5个anchor, 每个anchor属于背景和目标的类别的概率
4. 尺寸为(1, 256, 6, 6)的特征和尺寸为(1, 256, 22, 22)的特征, 送入box分支, 输出尺寸为(1, 20, 17, 17)的位置回归结果, 其中20表示5个anchor, 每个anchor的[x, y, w, h]偏移量

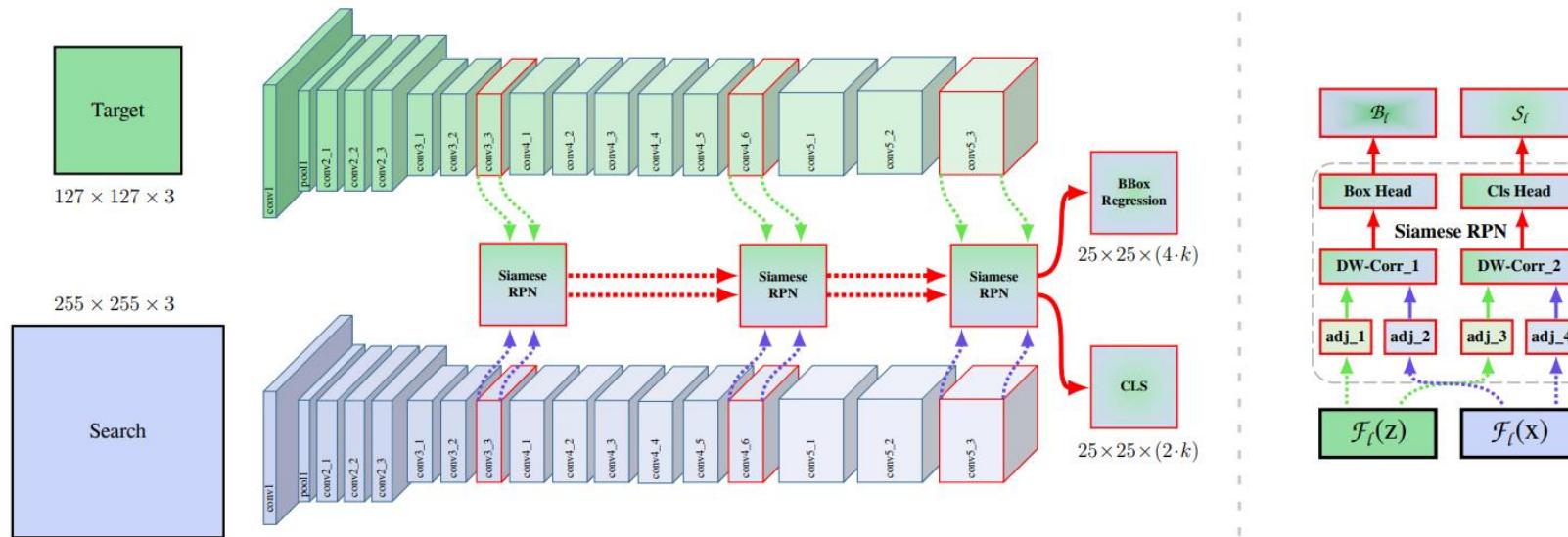
使用更深的网络, 没有取得更好的效果。最核心的原因是严格的平移不变性被破坏了

2. 经典方法——深度学习方法

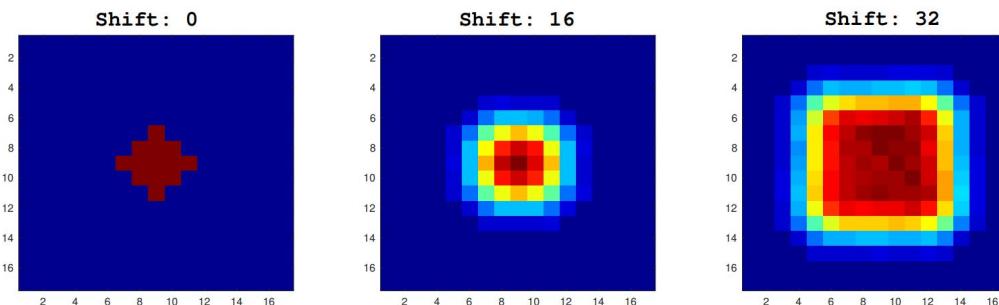
(5) Siamese based: SiameseRPN++



北京交通大学



- 提出了一种简单但是有效的采样策略来解决该问题，并且成功的训练了基于ResNet的跟踪器
- 提出了一种分层特征聚合的结构，用于互相关
- 提出了一种深度可分的相关结构来增强互相关系，从而产生与不同语义相关的多个相似图



$$f(z, x) = \varphi(z) * \varphi(x) + b$$

$$f(z, x') = f(x', z)$$

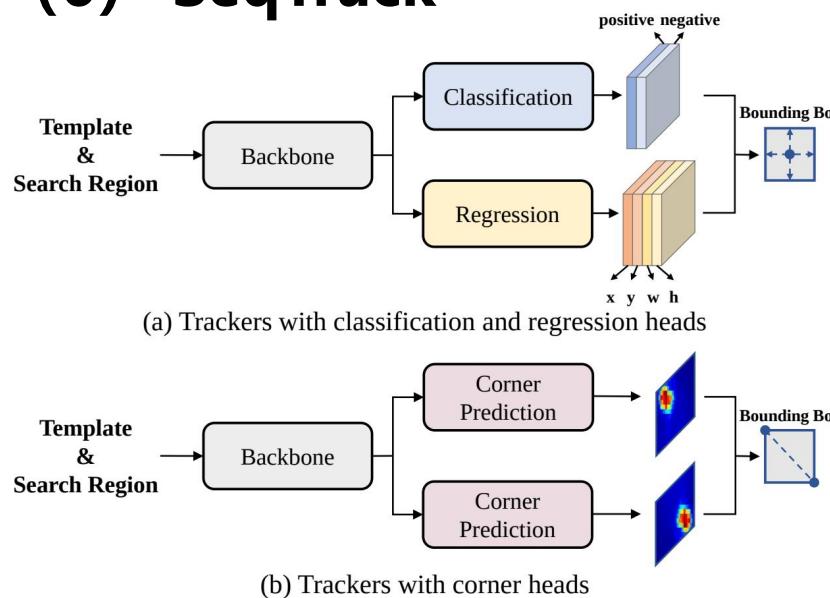
若违反平移不变性，则在特征图中，目标的中心可能就产生一定的偏移，即不再被体现在原图中目标的中心。

2. 经典方法——深度学习方法



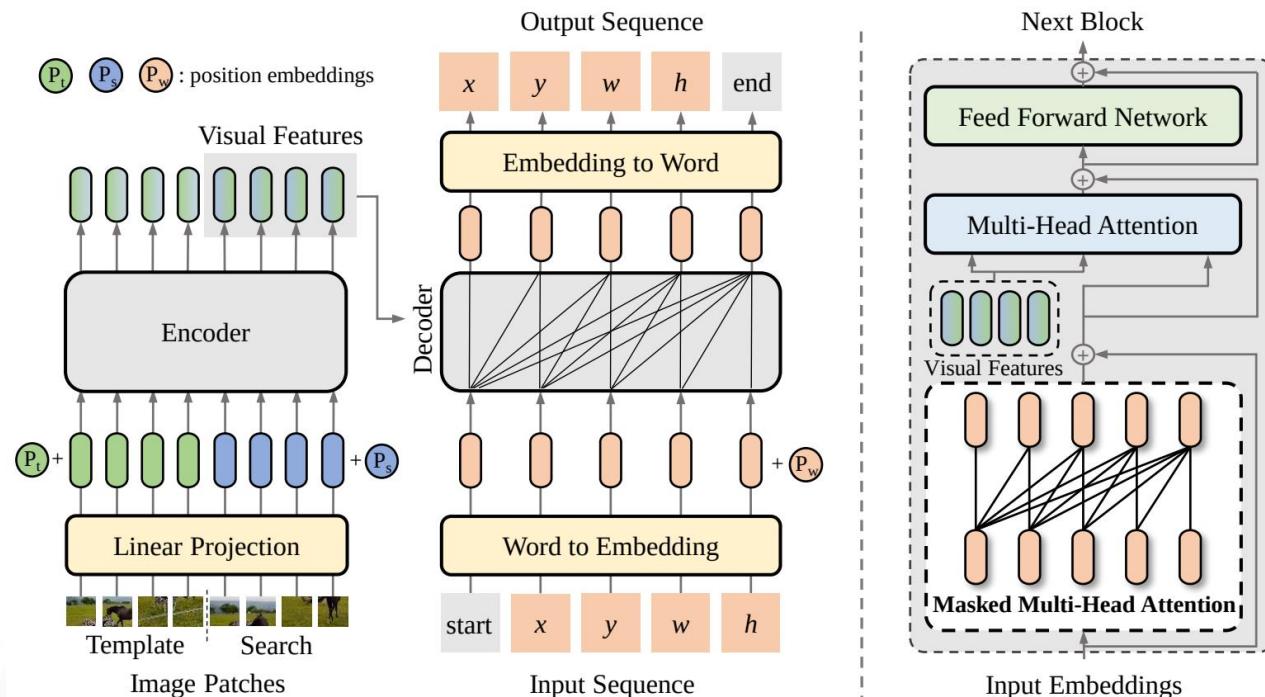
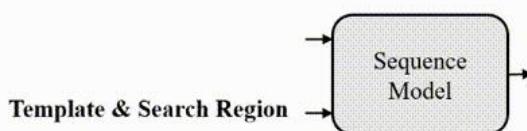
北京交通大学

(6) SeqTrack



模型复杂：每个子任务都需要一个定制化的预测头，导致框架变得复杂，不利于扩展

损失函数冗余：每个预测头需要一个或多个损失函数，引入额外超参数，使训练困难



SeqTrack采用了encoder-decoder形式的transformer

- Encoder提取Template与Search区域图片的视觉特征
- Decoder参考这些视觉特征，完成序列的生成

如果模型知道目标在图像中的位置，那么只需要简单地教模型读出目标边界框即可

采用自回归式的序列生成建模来解决目标跟踪任务，教模型把目标的位置作为一句话去“读”出来

3. 研究方向



北京交通大学

研究方向/分支

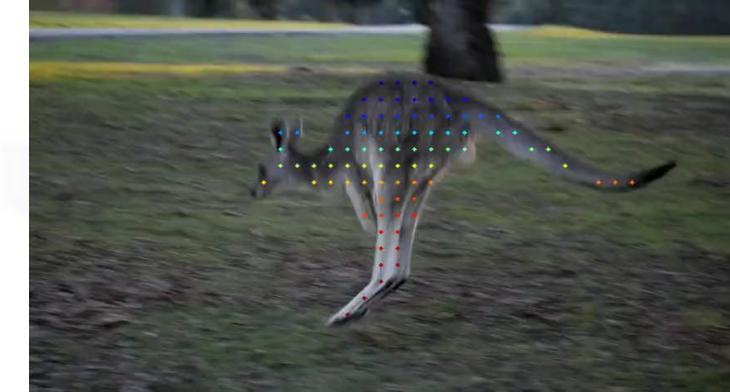
- 长期目标跟踪 (Long-term Tracking)
- 跨尺度目标跟踪 (Scale-aware Tracking)
- 目标姿态估计与跟踪 (Object Pose Estimation and Tracking)
- 目标跟踪与目标检测的融合 (Tracking and Detection Fusion)
- 在线学习与增量式目标跟踪 (Online Learning and Incremental Tracking)
- 跨模态目标跟踪 (Cross-modal Tracking)
- 立体目标跟踪 (Stereo Object Tracking)
- 真实场景目标跟踪 (Real-World Object Tracking)

3. 研究方向

Track Anything: Segment Anything Meets Videos



Tracking Everything Everywhere All at Once



4. 研究数据



北京交通大学

▲ 相关经典数据集举例：

(1) MOT Challenge: (<https://motchallenge.net/>)

- MOT Challenge是一个用于多目标跟踪研究的数据集，常见版本包括MOT16和MOT17。MOT16包含1,200多个视频序列，而MOT17扩展至14,000个视频序列，提供大规模多目标跟踪数据。



MOT17-03



MOT17-08



MOT17-07



MOT17-14

(2) UA-DETRAC: (<https://detrac-db.rit.albany.edu/download>)

- UA-DETRAC是用于城市自动驾驶和交通活动分析的数据集，包含高分辨率视频和详细目标标注信息。数据集覆盖各种城市交通场景，适用于自动驾驶系统研究和交通监控任务。



4. 研究数据



北京交通大学

▲ 相关经典数据集举例：

(3) KITTI-Tracking: (https://www.cvlibs.net/datasets/kitti/eval_tracking.php)

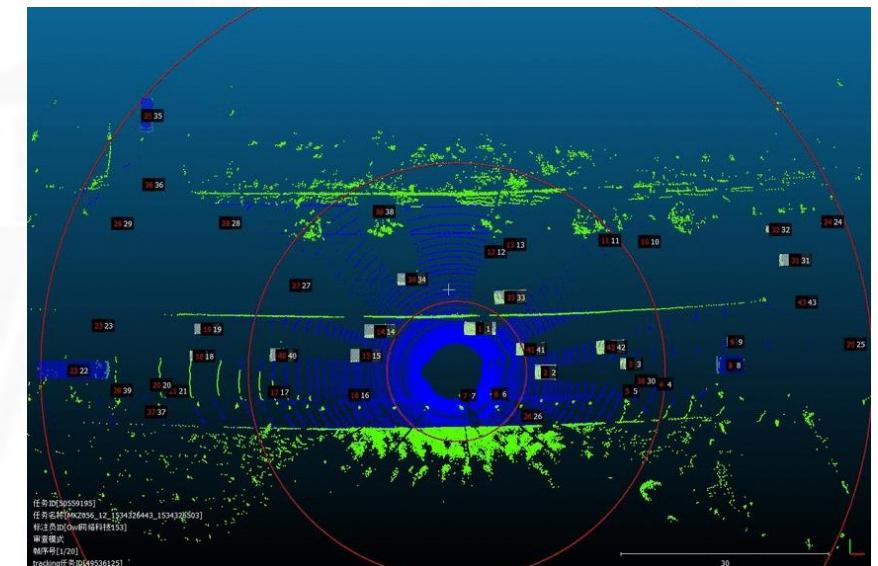
- KITTI-Tracking数据集是一个用于自动驾驶和多目标跟踪研究的数据集。它包含来自移动平台的高分辨率视频序列和激光雷达数据，涵盖各种交通场景，如城市道路、高速公路和乡村地区。数据集提供车辆、行人等目标的边界框和3D位置注释，用于目标检测和跟踪任务的评估和研究。



(4) APOLLOSCAPE: (<https://apolloscape.auto/tracking.html>)

- APOLLOSCAPE是一个用于自动驾驶研究的大规模综合数据集，包含丰富的高分辨率传感器数据，如激光雷达、相机和雷达。该数据集涵盖了城市、高速公路和郊区等多样场景，提供了车辆、行人等目标的详细注释。

APOLLOSCAPE旨在支持自动驾驶算法的训练和评估，为研究人员和开发者提供了丰富的实际驾驶数据。



4. 研究数据

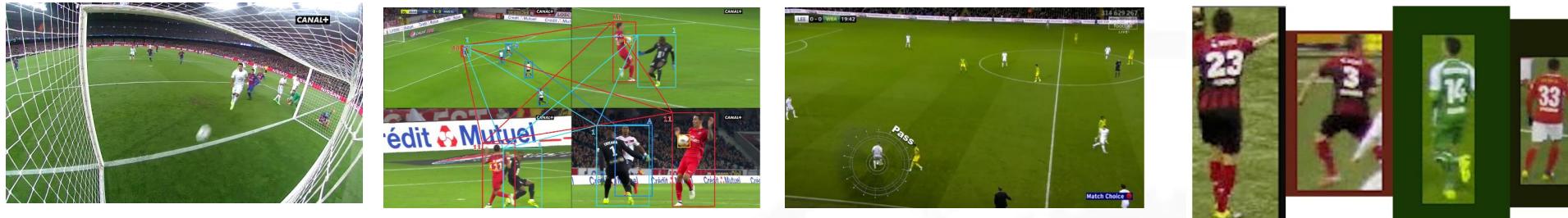


北京交通大学

▲ 相关经典数据集举例：

(5) SoccerNet Tracking: (<https://www.soccer-net.org/>)

- 专为足球比赛中球员追踪任务设计的数据集。它包含高质量的足球比赛视频，提供了球员的空间和时间注释，以支持目标追踪和分析。该数据集覆盖了多个比赛和场景，为研究足球比赛中运动轨迹和策略的算法提供了宝贵资源。



(6) SportsMOT: (<https://codalab.lisn.upsaclay.fr/competitions/12424>)

- 3个类别（即篮球、足球和排球）的240个视频片段组成。目标是仅跟踪操场上各种体育场景中的运动员（即，除了一些观众、裁判和教练）。



5. 评价指标



北京交通大学

▲ 常用的评价指标：(包括但不限于以下这些~)

- 召回率 (Recall, R)
- 精确度 (Precision, P)
- 准确度 (Accuracy, ACC)
- 假阳性率 (False Alarm Rate, FA)
- 漏报准确率 (Miss Accuracy, MA)
- 标识召回率 (IDentification Recall, IDR)
- 标识精度 (IDentification Precision, IDP)
- 标识F1分数 (IDentification F1-score, IDF1)
- 多目标跟踪准确率 (Multiple Object Tracking Accuracy, MOTA)
- 多目标跟踪精确度 (Multiple Object Tracking Precision, MOTP)

混淆矩阵		真实值	
		Positive	Negative
预测值	Positive	TP	FP (Type II)
	Negative	FN (Type I)	TN

- MOTA：用于计算在多目标跟踪问题中，跟踪器的输出与真实目标的位置之间的差异程度。

$$MOTA = 1 - \frac{(FN+FP+IDsw)}{GT}$$

- MOTP：用于计算基于跟踪系统的输出和真实目标的位置之间的距离误差。

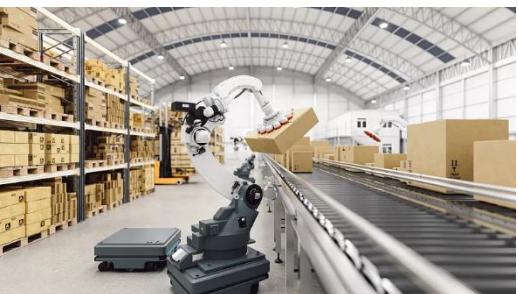
$$MOTP = \frac{\sum_t \sum_i d_{i,t}}{\sum_t c_t}$$

6. 应用领域

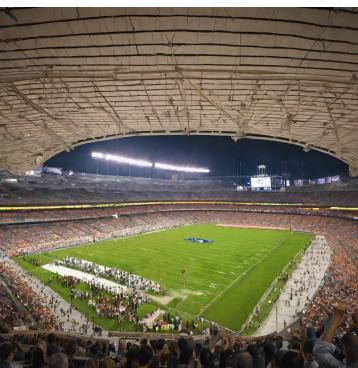
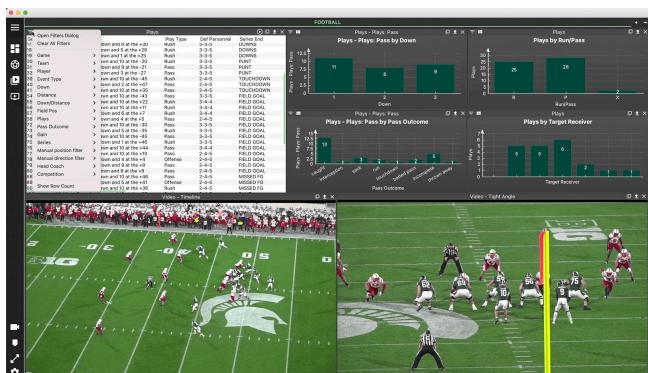
▲ 军事国防领域：边境监控与入侵检测、军事基地安全、舰船与飞机监控



▲ 工业生产与质量控制：生产线跟踪、缺陷检测、设备监控



▲ 体育赛事分析：运动员追踪、球场安全监测、比赛回放



6. 应用领域

▲ 智能监控与安防：人员追踪、异常行为检测、目标追踪



▲ 交通管理与车辆监测：车辆跟踪、交通事故分析、停车场管理



▲ 零售业智能分析：购物者追踪、商品关注度、排队分析



北京交通大学

7. 未来展望



北京交通大学

- **提高精度与效率：**

目标跟踪将不断追求更高的跟踪精度和更高的实时性能。通过引入更先进的**网络结构**、**特征提取**方法和**优化算法**，实现更准确、更快速的目标跟踪。

- **多模态数据融合**

目标跟踪将更关注于多模态数据融合。通过**结合图像、语音、传感器数据**等多种数据信息，可以更全面地感知和理解目标的运动和行为，提高跟踪的准确性和鲁棒性。

- **跨域自适应学习**

目标跟踪将更聚焦于跨域场景下的自适应学习。通过在**不同领域、不同视角**和**不同感知模态**之间进行知识迁移和特征共享，从而实现在新场景中的快速适应和准确跟踪。

- **改善跟踪实时性**

实时性是目标跟踪中的重要问题，在许多应用中具有重要的价值和需求，如自动驾驶、智能监控等。

目标跟踪将更关注**模型轻量化、高效计算、并行处理**和**快速数据处理**等方面，以提高跟踪的实时性和响应能力。

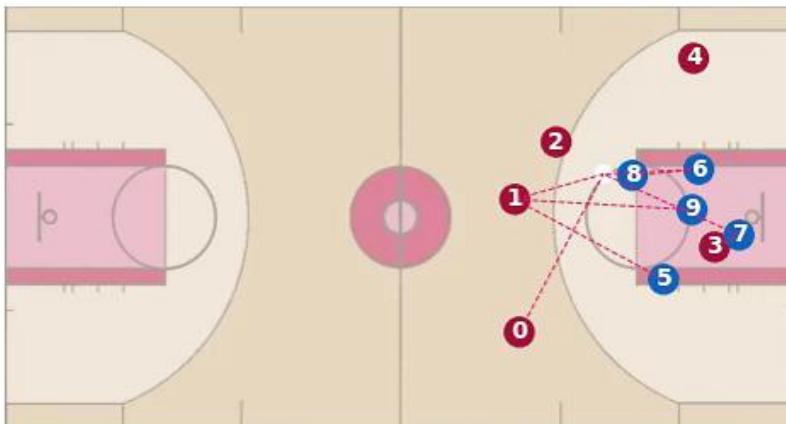


预见未来

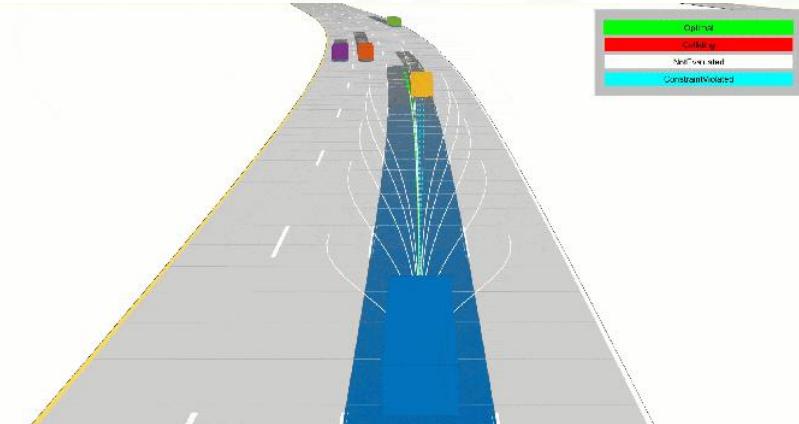


轨迹预测

Trajectory Prediction



Sport Scenarios



Driving Scenarios



Surveillance Scenarios



目录



1. 背景概述



2. 经典方法



3. 研究方向



4. 研究数据



5. 评价指标



6. 应用领域



7. 未来展望

1. 背景概述



北京交通大学

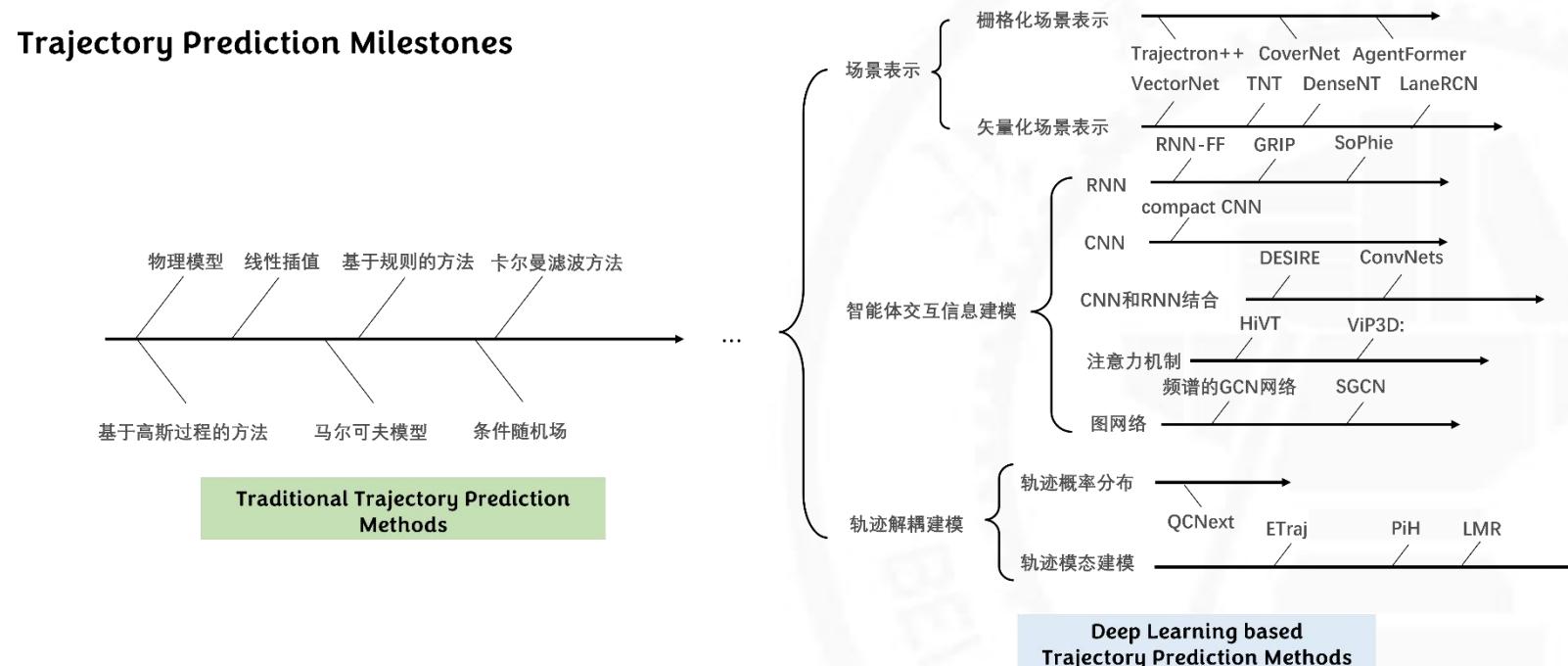
轨迹预测

▲ 基本定义：轨迹预测是指根据过去的观测数据和环境信息，预测目标在未来一段时间内的运动轨迹。在计算机视觉和机器人领域，轨迹预测是一项重要的任务，旨在推断目标的未来位置、速度和行为，以实现对目标的预测、规划和决策；

▲ 简单分类：基于传统方法和基于深度学习的方法；

▲ 研究意义：提供对目标未来运动的预测能力，为智能系统和应用提供决策、规划和交互的基础；

▲ 发展历程：



2. 经典方法-传统方法

(1) 基于卡尔曼滤波器

以目标追踪雷达为例

假设每5秒发起一次追踪

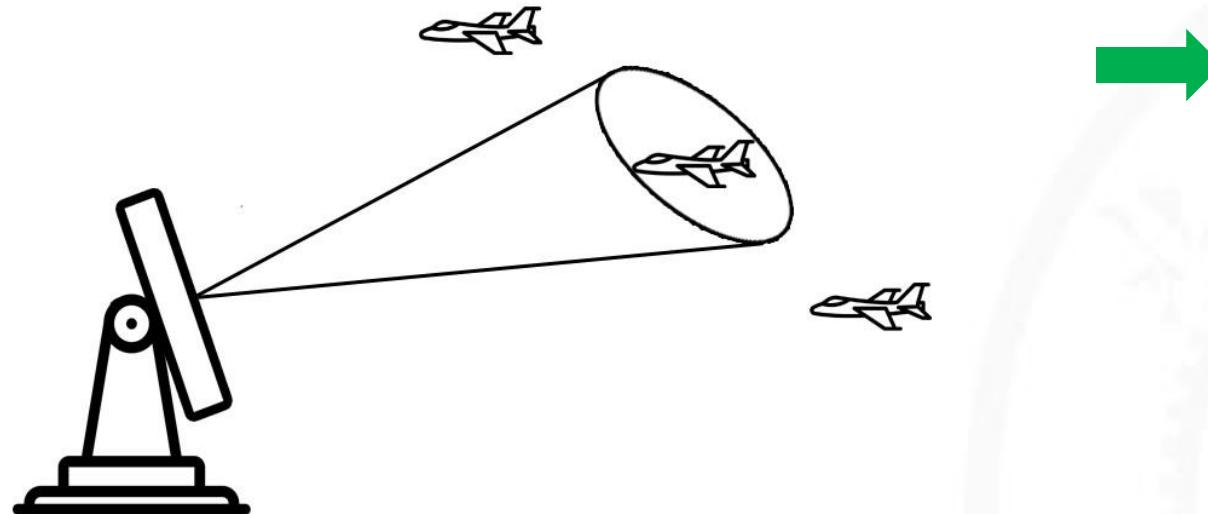
雷达波到达目标并返回后，开始估计目标当前的位置和速度。

同时预测目标在下一个测量周期即5秒后的位置。



北京交通大学

$$\begin{cases} x = x_0 + v_{x_0}\Delta t + \frac{1}{2}a_x\Delta t^2 \\ y = y_0 + v_{y_0}\Delta t + \frac{1}{2}a_y\Delta t^2 \\ z = z_0 + v_{z_0}\Delta t + \frac{1}{2}a_z\Delta t^2 \end{cases}$$



- 参数 $[x, y, z, v_x, v_y, v_z, a_x, a_y, a_z]$ 称为“**系统状态**”，当前状态作为输入，预测目标为下一个时刻的状态
- 运动方程称为**动态模型或状态空间模型**
- 测量的随机噪声或不确定性称为**测量噪声**，运动方程预测和实际目标的运动轨迹之间的差异称为**过程噪声**

为了提升雷达跟踪的精准度，一套能对抗**测量和模型的不确定性**的预测算法就至关重要。
最常用的跟踪和预测算法就是 **卡尔曼滤波**。

2. 经典方法-传统方法

(1) 基于卡尔曼滤波器

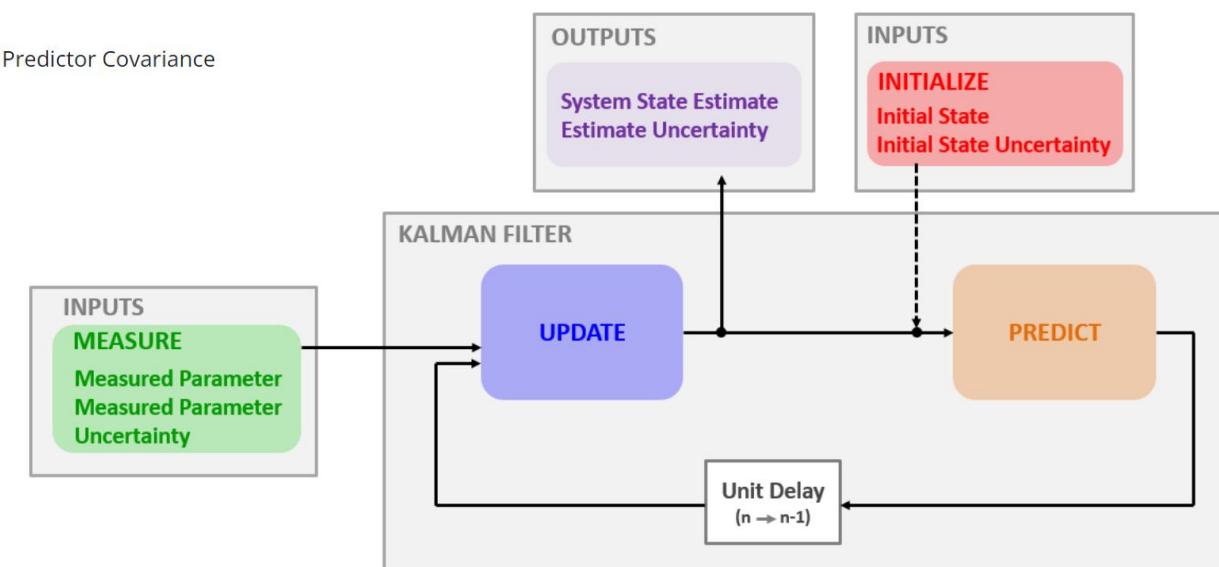


北京交通大学

方程	方程名	文献中可能出现的其他名字	
状态更新阶段	$K_n = \frac{p_{n,n-1}}{p_{n,n-1} + r_n}$	卡尔曼增益 Kalman Gain	权重方程 Weight Equation
	$\hat{x}_{n,n} = \hat{x}_{n,n-1} + K_n (z_n - \hat{x}_{n,n-1})$	状态更新 State Update	滤波方程 Filtering Equation
	$p_{n,n} = (1 - K_n) p_{n,n-1}$	协方差更新 Covariance Update	修正方程 Corrector Equation
状态预测阶段	$\hat{x}_{n+1,n} = \hat{x}_{n,n}$ (恒定动态模型)	预测器方程 Predictor Equation 转移方程 Transition Equation 预测方程 Prediction Equation 动态模型 Dynamic Model 状态空间模型 State Space Model	$\hat{x}_{n,n} = \frac{1}{n} \sum_{i=1}^n (z_i)$ 求平均公式: n 个测量的和再除以 n
	$\hat{x}_{n+1,n} = \hat{x}_{n,n} + \Delta t \dot{\hat{x}}_{n,n}$	状态外插 State Extrapolation	$= \frac{1}{n} \left(\sum_{i=1}^{n-1} (z_i) + z_n \right)$ 前 $n - 1$ 个测量的和再加上最近一次的测量值整体除以 n
	$\hat{x}_{n+1,n} = \hat{x}_{n,n}$ (速度恒定动态模型)		$= \frac{1}{n} \sum_{i=1}^{n-1} (z_i) + \frac{1}{n} z_n$ $\frac{1}{n}$ 乘进去展开
	$p_{n+1,n} = p_{n,n}$ (恒定动态模型)		$= \frac{1}{n} \frac{n-1}{n-1} \sum_{i=1}^{n-1} (z_i) + \frac{1}{n} z_n$ 给求和项同时乘以并除以 $n - 1$
	$p_{n+1,n}^x = p_{n,n}^x + \Delta t^2 \cdot p_{n,n}^v$	协方差外插 Covariance Extrapolation	$= \frac{n-1}{n} \frac{1}{n-1} \sum_{i=1}^{n-1} (z_i) + \frac{1}{n} z_n$ 调整顺序 橘黄色的项就是上一时刻的估计值
	$p_{n+1,n}^v = p_{n,n}^v$ (速度恒定动态模型)		$= \frac{n-1}{n} \hat{x}_{n-1,n-1} + \frac{1}{n} z_n$ 把求和项用上一时刻的估计值替换掉

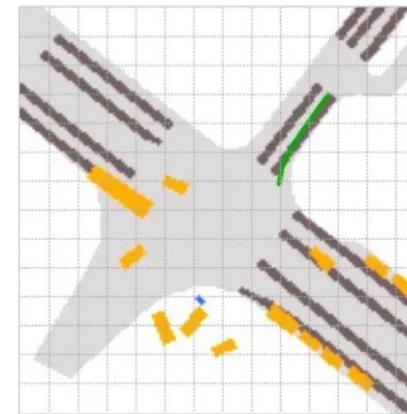
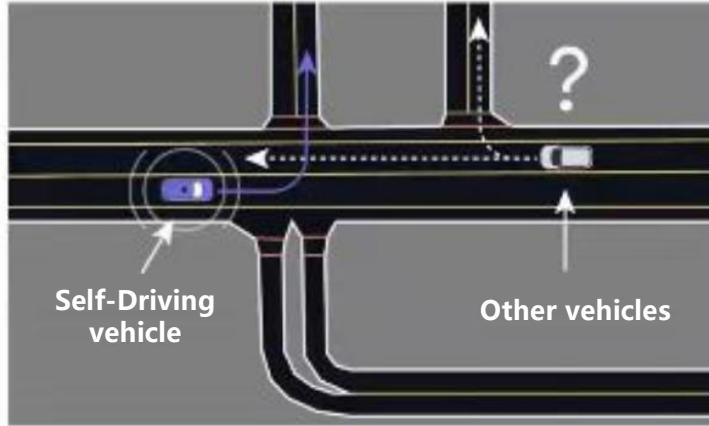


如何给金条称重?



2. 经典方法-深度学习方法

(1) CoverNet: 场景表示



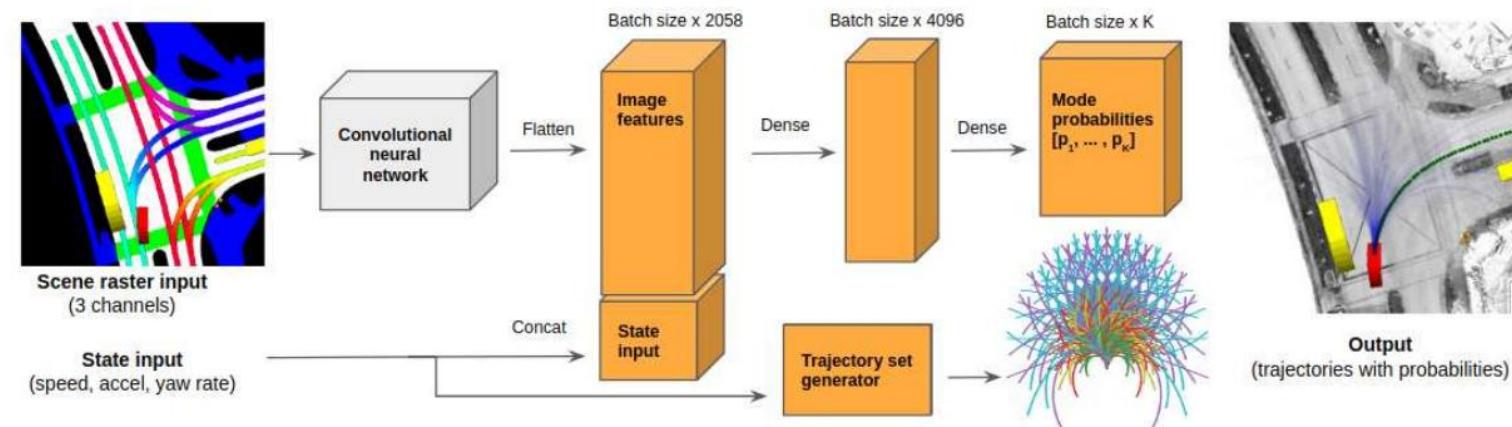
以**目标为中心的3D目标检测、多目标跟踪和轨迹预测**是目前主流的感知模块

以**栅格为中心的感知**被认为是一个很有前途的解决方案，因为它可以在**不知道物体的情况下**提供**三维周围空间中任何位置的占据和运动**

$$p(s_F|X) = \int_{\tau \in \mathcal{T}(c_p)} p(\tau|X)p(s_F|\tau, X)d\tau$$

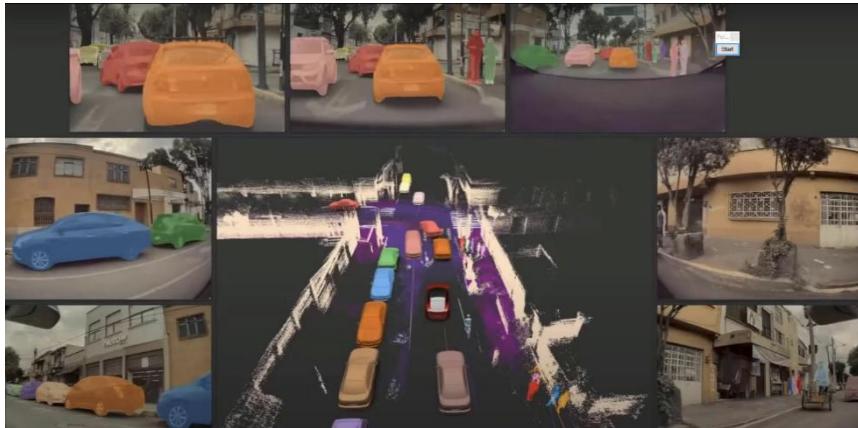
$$p(s_F|X) = \sum_{\tau \in \mathcal{T}(c_p)} p(\tau|X)p(s_F|\tau, X)$$

- s_F 表示target Agent在未来的行为
- X 表示过去到当前时刻的交通场景信息
- τ 代表不同的模态

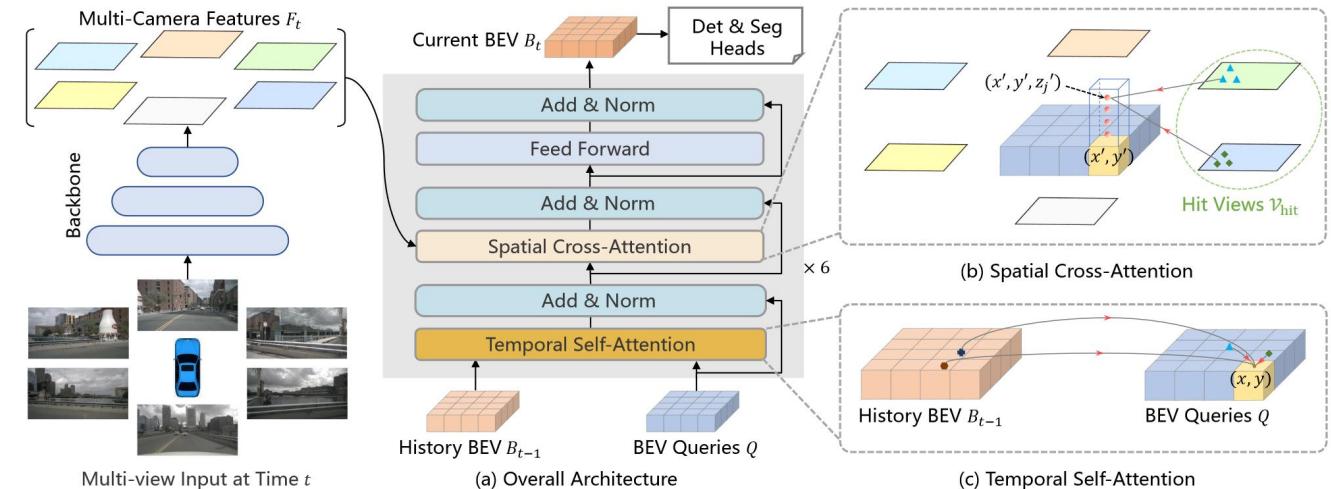


2. 经典方法-深度学习方法

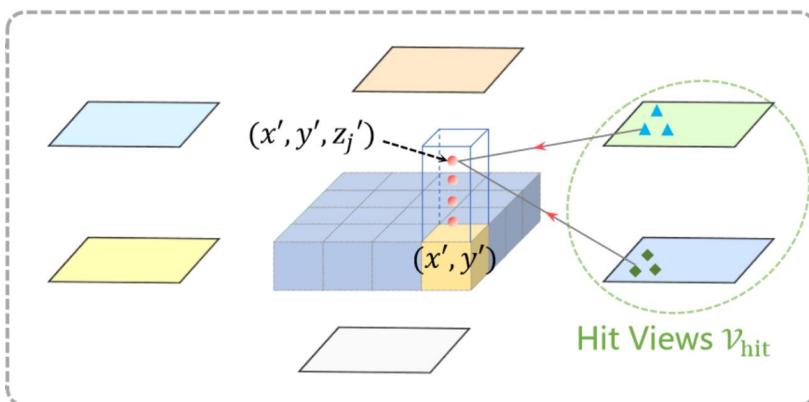
(2) BevFormer: 场景表示



无人系统中，感知模块需要一个上帝视角



假如对于 BEV 平面上的 (x, y) 处有一个智能体，它所对应的特征应该由多个位置的视觉传感器特征汇聚而成，将 BEV 坐标系下的三维点映射回图像平面后可以去查询并融合更加准确的特征；



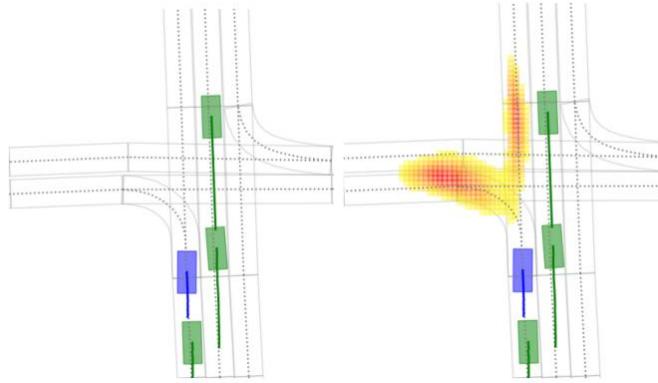
1. 要融合多个视角的特征相当于query会遍历所有视角找有用的信息
2. 3D空间的点要投影到2D空间，但是由于遮挡或者相机内外参不准
3. 那么投影点就是一个参考而已，在这个基础上附近再进行特征采样

2. 经典方法-深度学习方法

(3) HOME: 场景表示

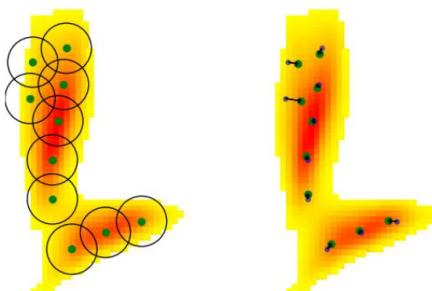


北京交通大学



- target agent (blue)
- neighbor (green) trajectories
- Context map

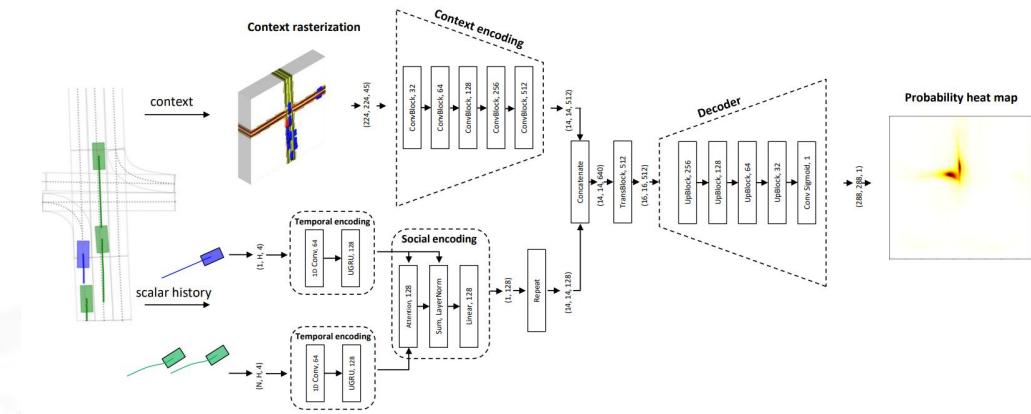
高清地图光栅化为5个语义通道: **drivable area**通道, **lane boundaries**通道, 和**编码表示带运动方向directed center-lines**的HSV-3通道; 另外加上**目标代理轨迹表示的运动长方形**20个历史通道以及**其他代理的20个通道**分辨率**为0.5m/pixel**。Agent的**标量历史编码**, 作为输入是一组2D坐标。



(a) MR sampling

(b) FDE sampling

- 训练过程, 需要准备一张**真值的heatmap图**, 这张**heatmap**是由**真值终点为中心点, 按照高斯分布 (标准差为4pixel)**生成的。
- 得到**heatmap**之后, 需要设计相应的**sampling**算法得到K个**end points**。
- 使用**双线性插值**得到了**0.25m/pixel**的分辨率, 以得到更为精确的位置坐标

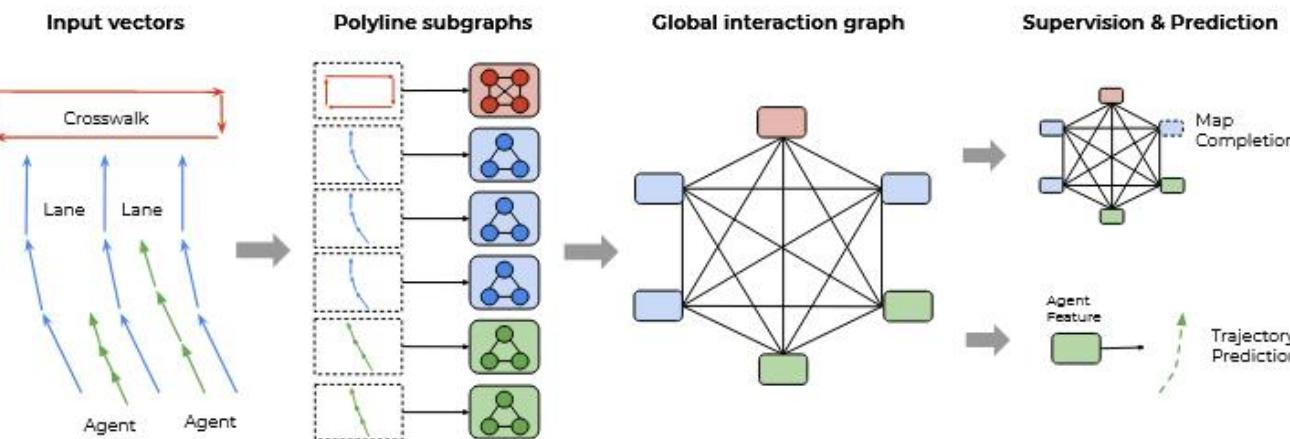


用渲染的**bev**图作为环境信息输入, concat上周围**车辆的轨迹信息**, 最终decoder出来一个**heatmap**图

2. 经典方法-深度学习方法

(4) VectorNet: 场景表示

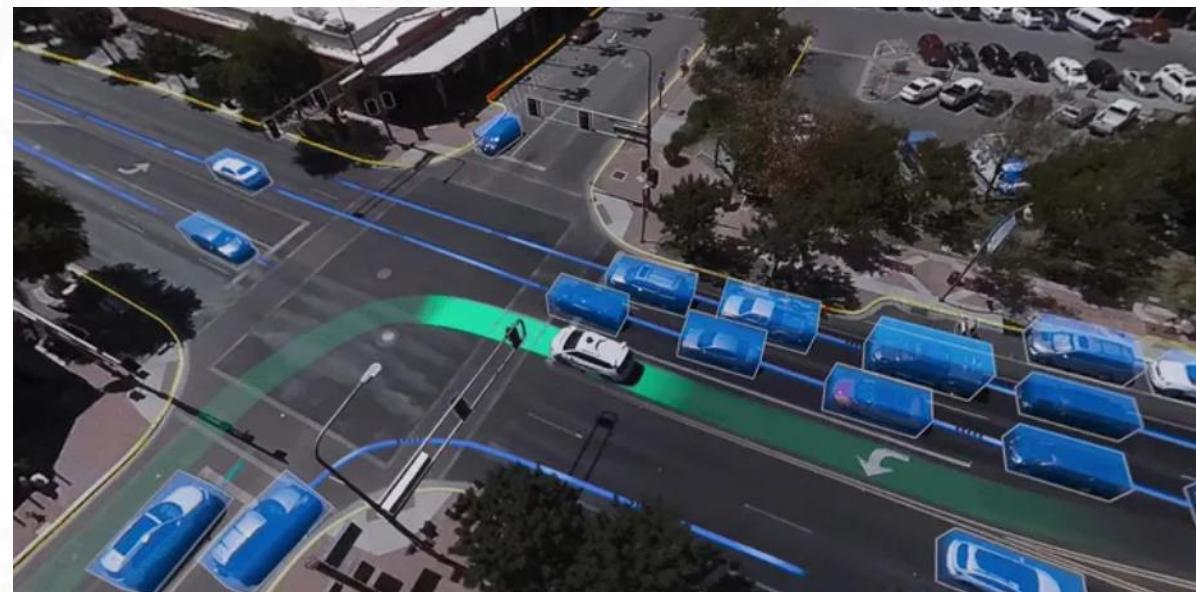
- **information loss:** 有些信息无法渲染 (刹车, 左转等)
- **hard to capture topology information,** 如车道间的连通性, 车道之间的依赖关系
- **high CPU consumption:** 栅格化渲染耗费的 cpu 资源大



Representing trajectories and maps

要表示的主要有两类信息：车轨迹 和 地图(车道线 人行道等)，都可视为折线，多边形可视为闭合折线

- 轨迹、车道线都被稀疏采样成线段
- 多边形也被稀疏采样成线段
- 这些等距采样的线段构成一条条折线 polyline
- 一个 Polyline 由多个线段组成，线段 v_i 由以下几个方面组成

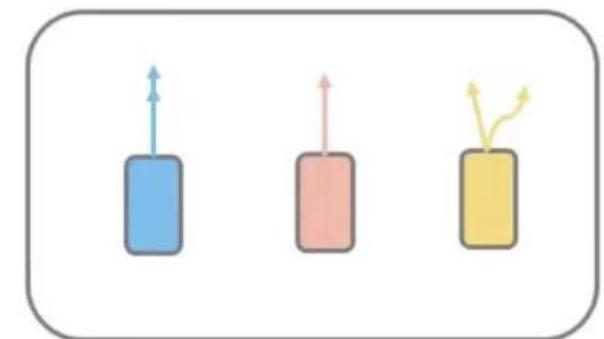
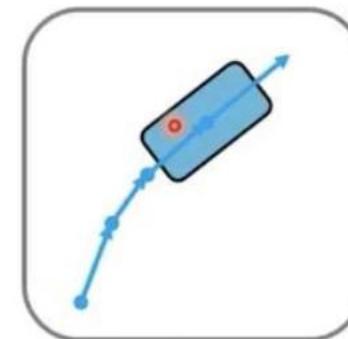
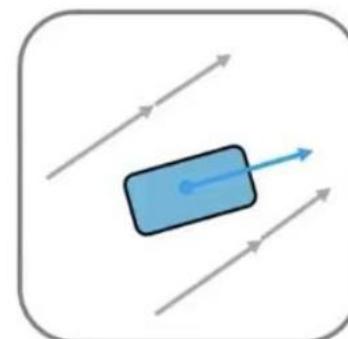
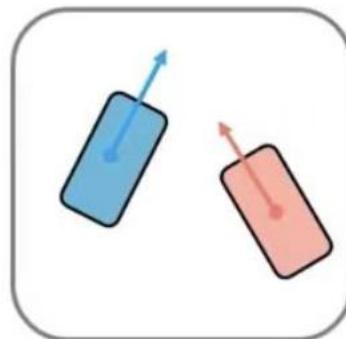


2. 经典方法-深度学习方法



北京交通大学

- Agent-Agent Interaction (A-A)
 - 学习智能体之间的关系
- Temporal Dependency
 - 捕捉每个智能体的时序信息
- Agent-Lane Interaction (A-L)
 - 学习智能体和车道段之间的关系
- Multi-modal Future Output
 - 每个智能体的多模态轨迹输出

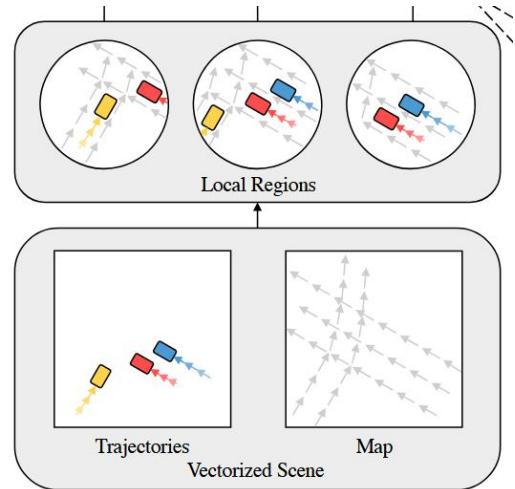


2. 经典方法-深度学习方法

(5) HiVT & QCNet: 场景表示



北京交通大学



空间不变性-HiVT

$$\{p_i^t - p_i^{t-1}\}_{t=1}^T$$

智能体

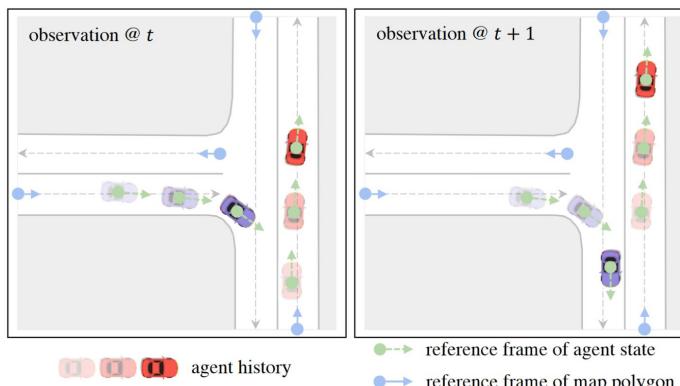
$$p_\xi^1 - p_\xi^0$$

车道
线

其中, $p_i^t \in \mathbb{R}^2$ 是智能体 i 在时间步 t 处的坐标位置, T 是总的时间长度; 车道段为 ξ , $p_\xi^1 \in \mathbb{R}^2$, $p_\xi^0 \in \mathbb{R}^2$ 分别为车道线的起点和终点。

将空间划分为 N 个局部区域, 每个局部区域以场景中的一个智能体为中心

空间不变性-QCNet



$$(p_i^t, \theta_i^t, t)$$

$$(p_i^s, \theta_i^s, s)$$

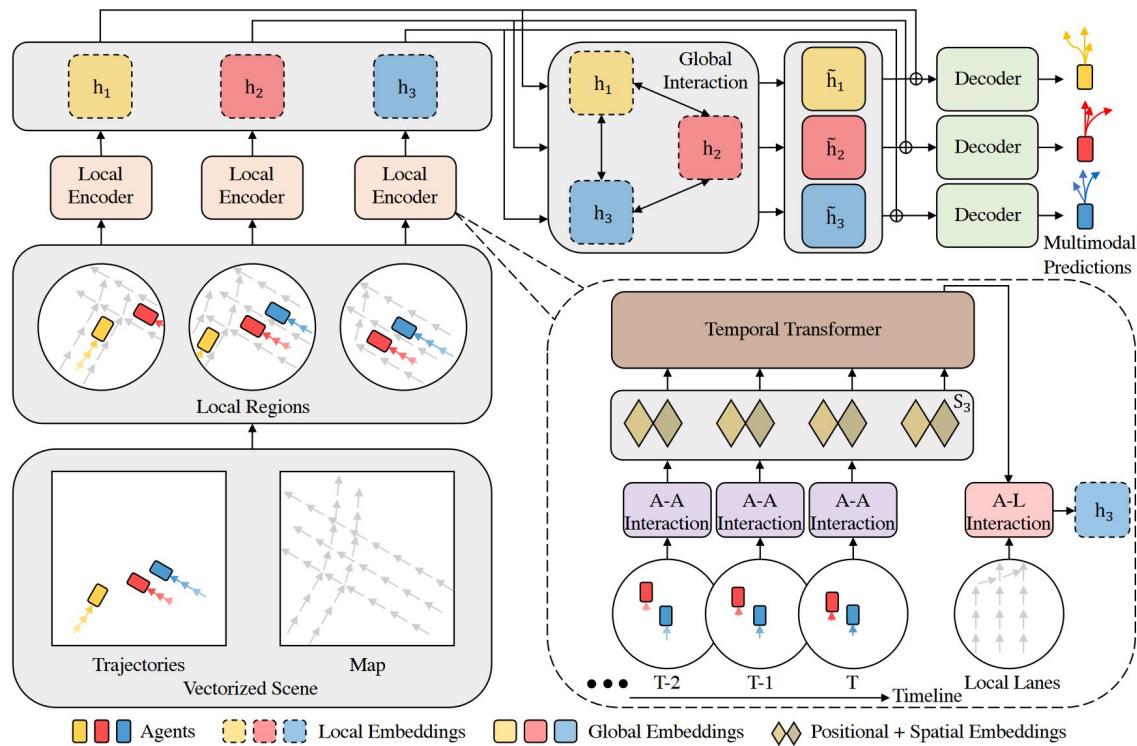
- **Relative distance:** $\|p_i^s - p_i^t\|_2$
- **Relative direction:** $\text{atan2}(p_{i,y}^s - p_{i,y}^t, p_{i,x}^s - p_{i,x}^t) - \theta_i^t$
- **Relative orientation:** $\theta_i^s - \theta_i^t$
- **Time gap** $s - t$
- 将 4D 描述符转化为傅里叶特征, 并将它们传递给 MLP 以生成relative positional embedding $r_{j \rightarrow i}^{s \rightarrow t}$

2. 经典方法-深度学习方法



北京交通大学

(6) HiVT: 智能体交互信息建模



- 局部编码器

通过建模智能体之间的相互作用、时间依赖性和智能体与道路的交互来捕捉局部信息

- 全局交互模块

用于捕捉场景中的长程依赖关系

- 多模态未来解码器

来预测智能体的未来运动

$$f\left(\{p_i^t\}_{t=1}^{T'}\right) = \sum_{k=1}^K \pi_{i,k} \prod_{t=1}^{T'} Laplace(p_i^t | \mu_{i,k}^t, b_{i,k}^t)$$

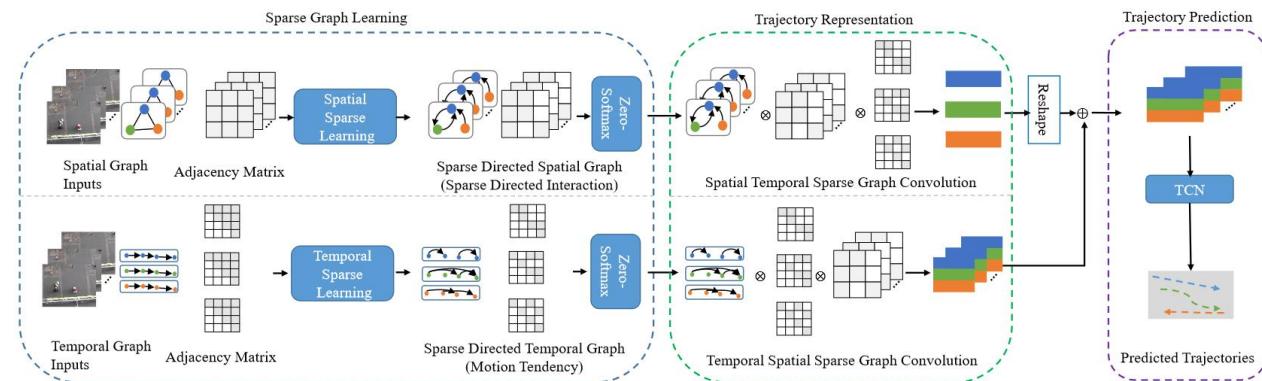
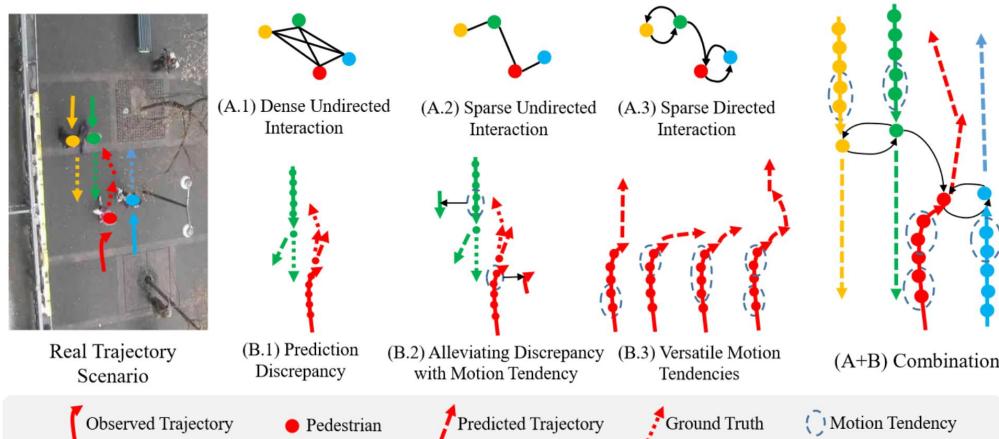
其中， $\{\pi_{i,k}\}_{k=1}^K$ 是混合系数，第 k 个混合分量在时间步 t 的拉普拉斯密度由位置 $\mu_{i,k}^t$ 和尺度 $b_{i,k}^t$ 参数化。

2. 经典方法-深度学习方法

(7) SGCN: 智能体交互信息建模



北京交通大学



假设轨迹 (x_n^t, y_n^t) 服从bi-variate Gaussian分布, $\mathcal{N}(\hat{\mu}_n^t, \hat{\sigma}_n^t, \hat{\rho}_n^t)$, 其中 $\hat{\mu}_n^t$ 为均值, $\hat{\sigma}_n^t$ 为标准差, $\hat{\rho}_n^t$ 为相关系数。

$$L^n(\mathbf{W}) = - \sum_{t=T_{\text{obs}}+1}^{T_{\text{pred}}} \log P((x_n^t, y_n^t) | \hat{\mu}_n^t, \hat{\sigma}_n^t, \hat{\rho}_n^t)$$

智能体之间的交互:

- 稠密
- 无向

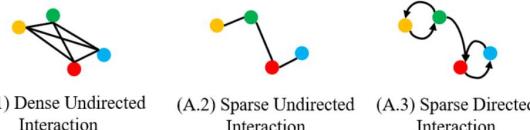
Sparse Directed Spatial Graph

$$E_{\text{spa}} = \phi(G_{\text{spa}}, W_E^{\text{spa}}),$$

$$Q_{\text{spa}} = \phi(E_{\text{spa}}, W_Q^{\text{spa}}),$$

$$K_{\text{spa}} = \phi(E_{\text{spa}}, W_K^{\text{spa}}),$$

$$R_{\text{spa}} = \text{Softmax}\left(\frac{Q_{\text{spa}} K_{\text{spa}}^T}{\sqrt{d_{\text{spa}}}}\right),$$



- 通过稀疏空间和时间图卷积网络从稀疏有向空间图和稀疏有向时间图中提取特征
- 通过TCN估计未来轨迹点的双高斯分布参数
- 能够有效地捕捉行人之间的交互和运动趋势, 通过建模稀疏有向图来提高轨迹预测的准确性

$$A_{\text{spa}} = (M_{\text{spa}} + I) \odot R_{\text{spa}}^{\text{s-t}}$$

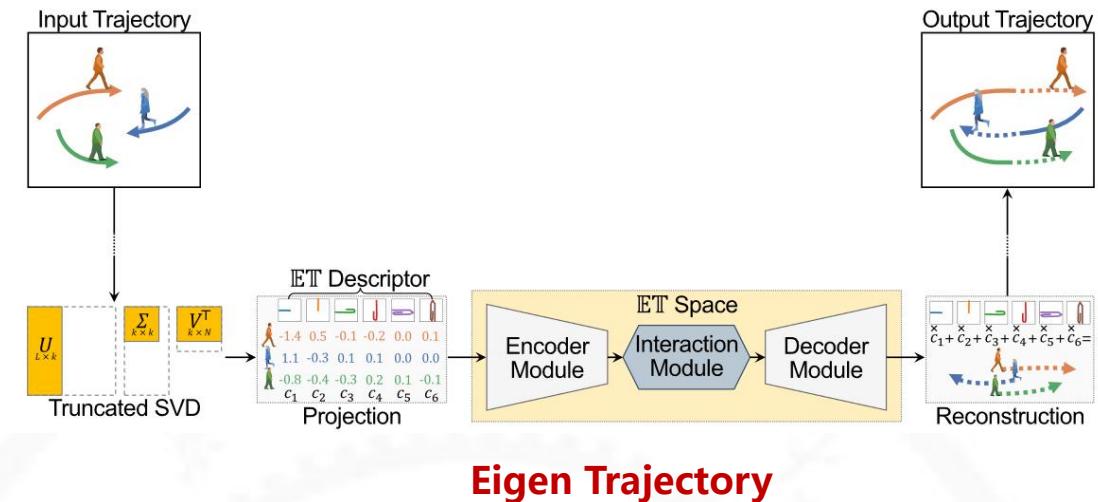
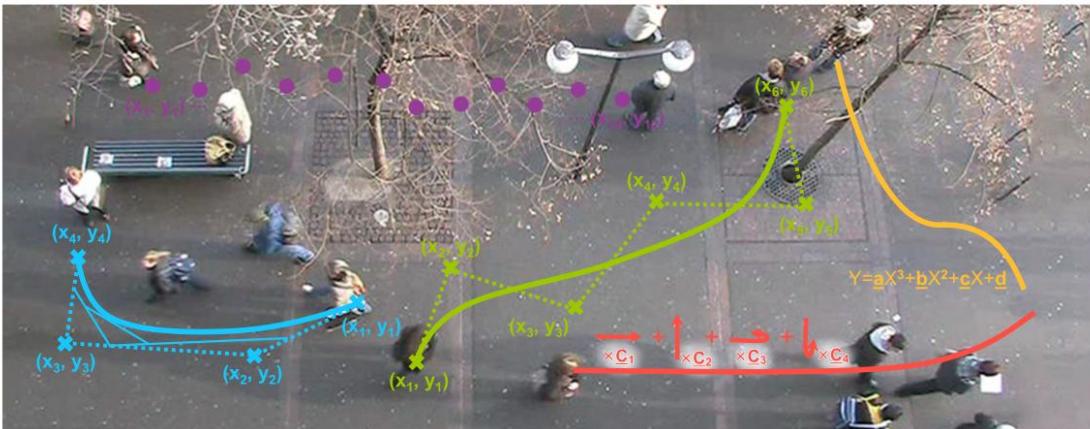
2. 经典方法-深度学习方法



北京交通大学

(8) EigenTrajectory: 轨迹解耦建模

- Raw point
- Bézier curve
- B-spline
- Polynomial curve
- EigenTrajectory

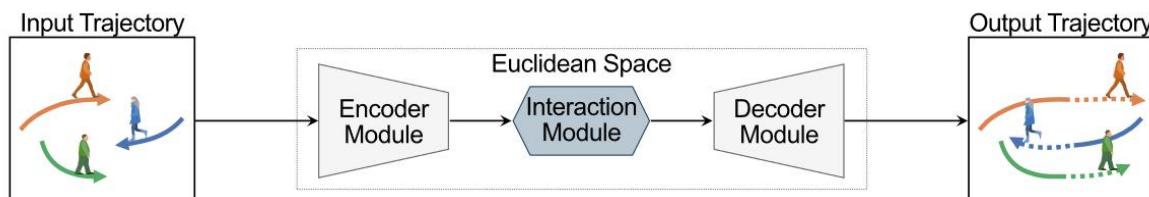


Eigen Trajectory

$$A = \{(x_n^t, y_n^t) | t \in [1, \dots, T_{obs}]\} \quad A = U_{obs} \Sigma_{obs} V_{obs}^T$$

轨迹的表示方法：

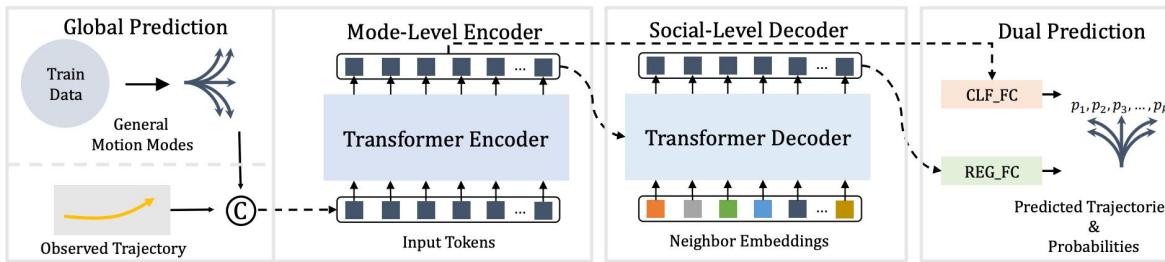
- Bezier curve
- B-spline
- Polynomial curve



- u1 和 u2 分别编码 x 和 y 方向上的等速运动。
- u3和u4分别表示x轴和y轴方向上的速度变化。
- 在 u5 和 u6 中通过将它们与先前的特征向量相结合来表示更详细的轨迹。

2. 经典方法-深度学习方法

(9) TUTR: 轨迹分模态建模



首先对所有的轨迹进行标准化，然后聚类。

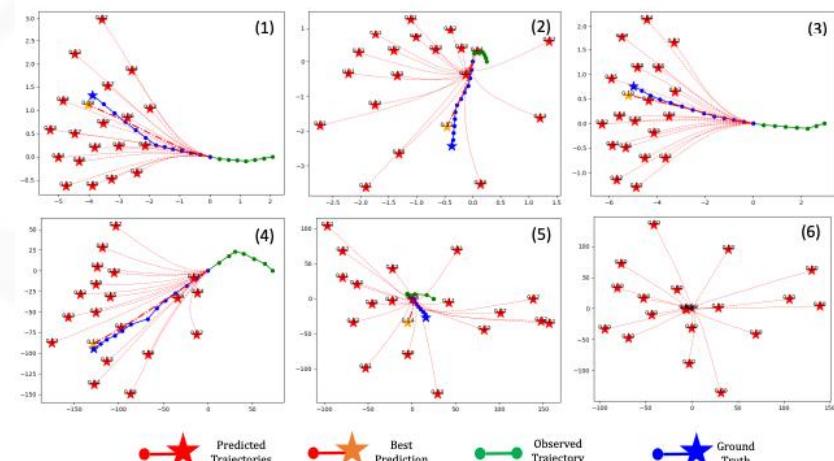
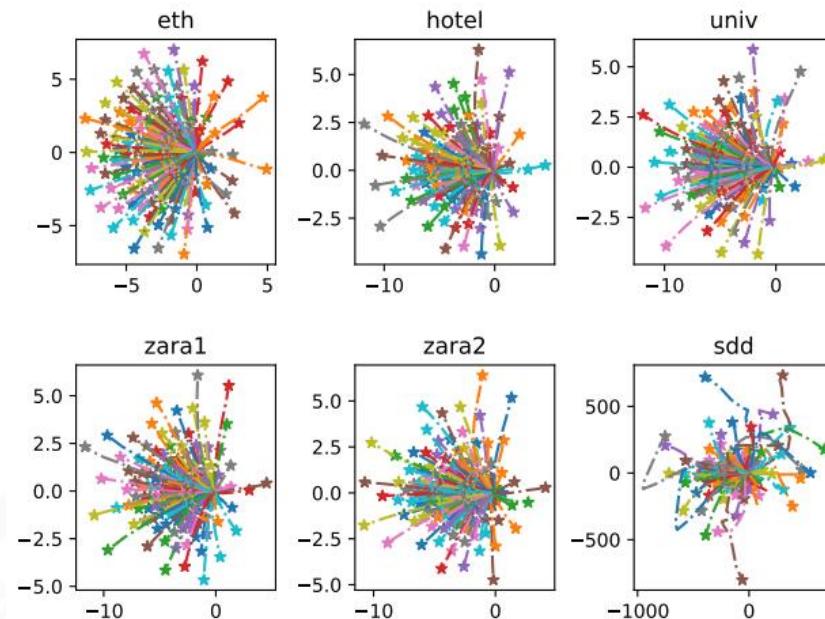
$$\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_L\} \in \mathbb{R}^{L \times T_{pred} \times 2}$$

轨迹进入模型预测的时候，将聚类中心作为固定的
模态直接进行显式编码：

$$E_c = \emptyset(\mathbf{C}, \mathbf{W}_c)$$

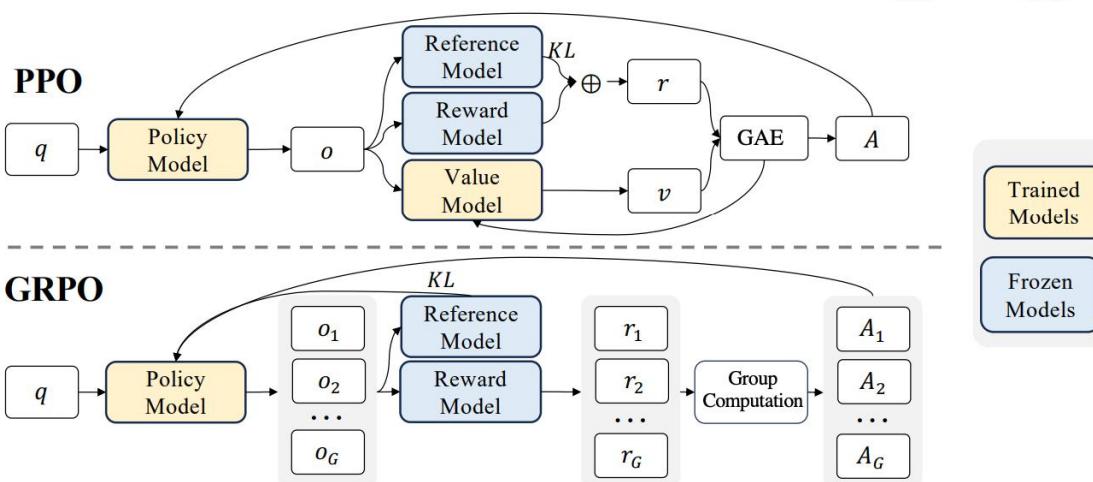
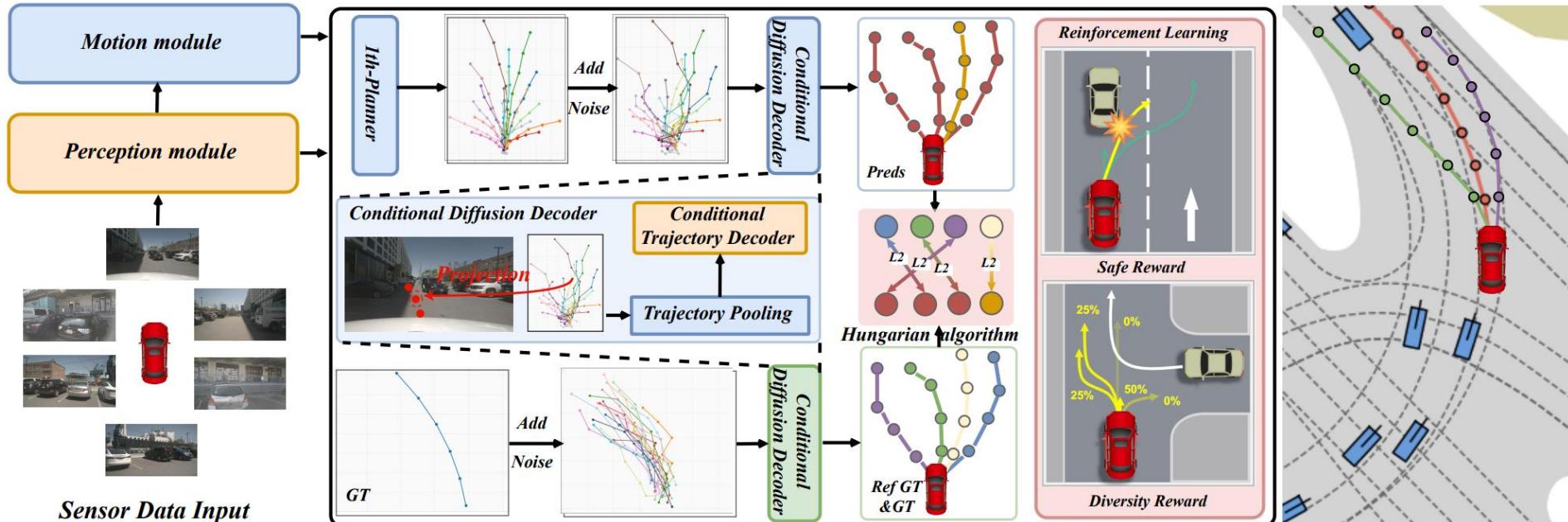
$$E_o = \emptyset(\mathbf{X}, \mathbf{W}_c)$$

$$E_e = E_c + E_o$$



2. 经典方法-深度学习方法

(10) DRIVE: Diffusion轨迹分模态建模+强化学习



$$r_{\text{div}} = \frac{2}{M(M-1)} \sum_{i=1}^M \sum_{j=i+1}^M \|\hat{\tau}^{(i)} - \hat{\tau}^{(j)}\|_2$$

$$r_{\text{safe}}(\hat{\tau}) = -\frac{1}{T} \sum_{t=1}^T \mathbb{I}[D_{\text{safe}}(\hat{x} * t) < d * \text{thresh}] ,$$

$$r(\hat{\tau}^{(m)}) = r_{\text{div}}(\hat{\tau}^{(m)}) + \lambda_{\text{safe}} \cdot r_{\text{safe}}(\hat{\tau}^{(m)}).$$

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q, \{o_i\} \sim \pi_{\theta_{old}}} \left[\frac{1}{G} \sum_{i=1}^G \mathcal{L}_i - \beta \mathbb{D}_{KL}(\pi_\theta \| \pi_{\text{ref}}) \right] , \quad (16)$$

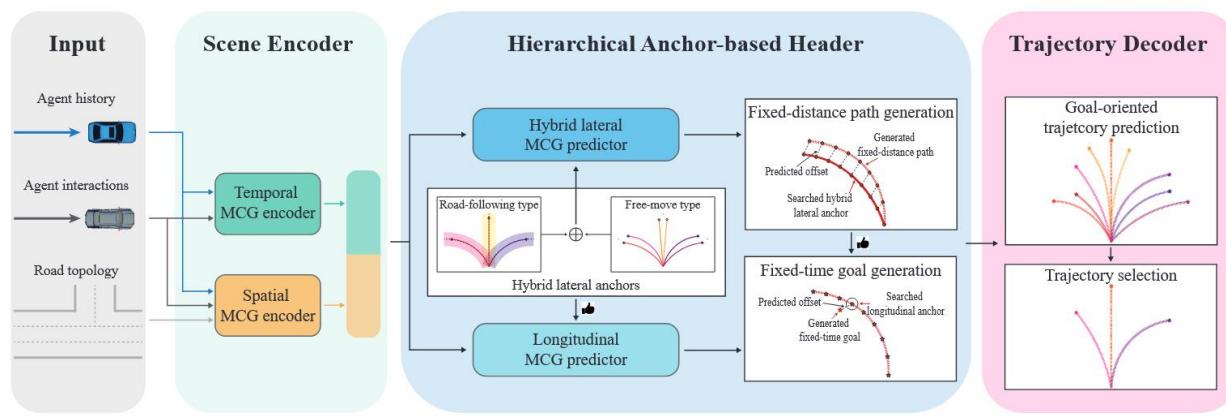
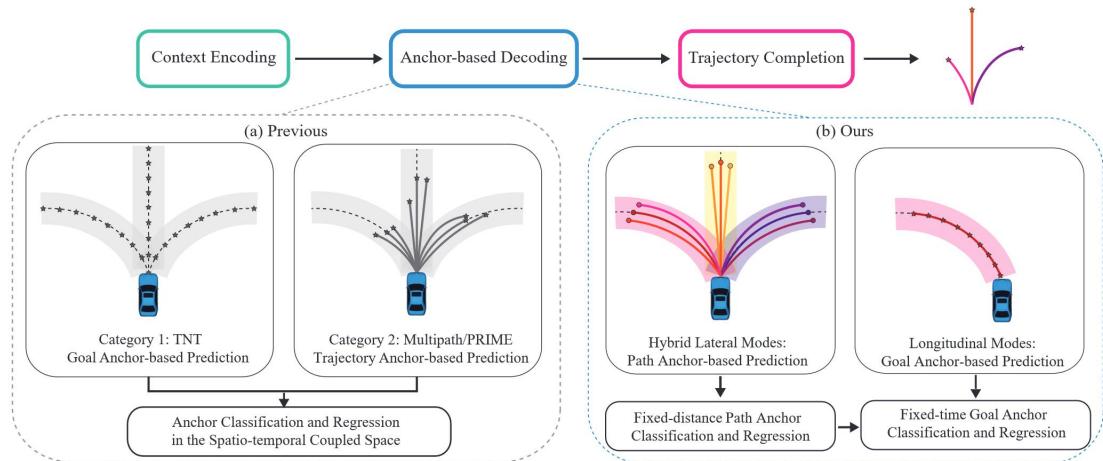
$$\mathcal{L}_i = \min(w_i A_i, \text{clip}(w_i, 1-\epsilon, 1+\epsilon) A_i) , \quad (17)$$

2. 经典方法-深度学习方法

(11) PiH: 轨迹解耦建模



北京交通大学



轨迹的表示方法

- 将未来意图分解为横向固定距离路径和纵向固定时间
- 横向固定距离路径负责“去哪里”问题
- 纵向固定时间负责“什么时候到达”问题

PiH 的输出是在固定的时间步长预测范围内目标车辆的一组 K 条轨迹, 每一个轨迹有两个维度, 包括横向路径 (“where to go”) 和纵向速度 (“when will it arrives”)。

分层预测策略以概率预测 L 条固定距离路径, 然后为每个横向模式预测 M 个固定时间目标作为纵向模式。

$$N = L \cdot M$$

$$p(Y|X) = \sum_{i=1}^L \sum_{j=1}^M \underbrace{p(\mathcal{P}_i|X)}_{lateral} \underbrace{p(g_i^j|\mathcal{P}_i, X)}_{longitudinal} p(Y|g_i^j, \mathcal{P}_i, X)$$

3. 研究方向



北京交通大学

研究方向/分支

- 融合多模态数据的轨迹预测方法
- 考虑目标行为意图的轨迹预测研究
- 跨域和迁移学习在轨迹预测中的应用
- 结合环境感知信息进行轨迹预测的研究
- 考虑不确定性和可解释性的轨迹预测方法
- 强化学习在轨迹预测中的应用
- 轨迹预测中的异常检测和异常轨迹识别
- 轨迹预测的实时性和效率优化

4. 研究数据

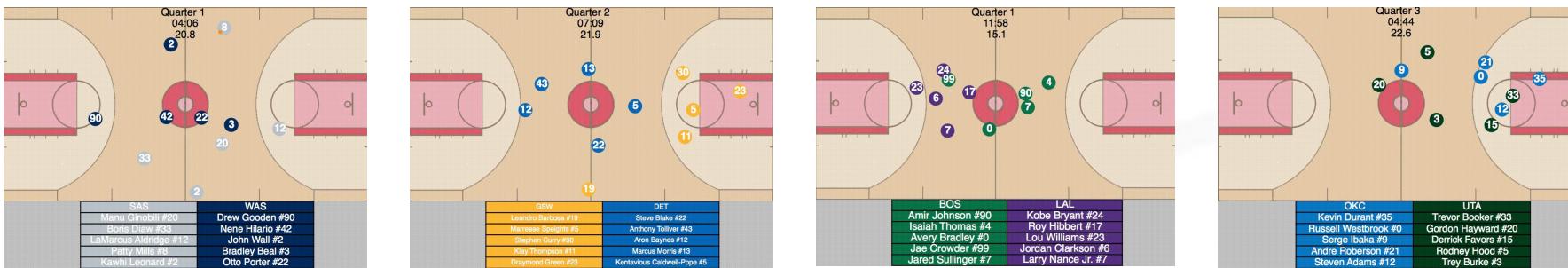


北京交通大学

▲ 相关经典数据集举例：

(1) NBA Player Movements: (<https://github.com/linouk23/NBA-Player-Movements>)

- 用于根据原始 SportVU 日志可视化 NBA 比赛的脚本。



(2) Fitogther data & Metrica data: (<https://github.com/pientist/ballradar>)

- Fitogther 数据：15 条匹配的 GPS 追踪数据与 Fitogther 提供的相应事件数据相结合。
- Metrica 数据：Metrica Sports 提供的 3 场公开光学追踪和赛事数据。



4. 研究数据

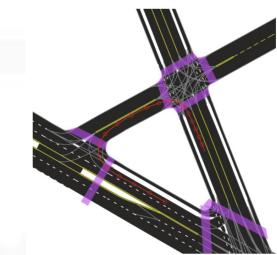
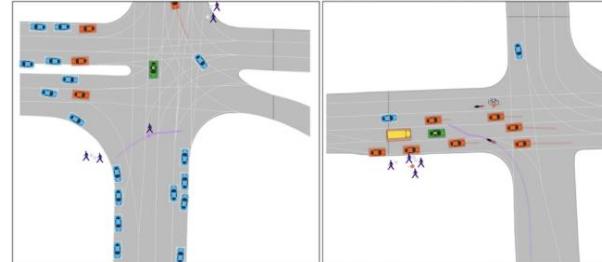


北京交通大学

▲ 相关经典数据集举例：

(3) Argoverse: (<https://www.argoverse.org/>)

- Argoverse 包括两个运动预测数据集：原始的 Argoverse 1 运动预测数据集及其后继者 Argoverse 2 运动预测数据集，后者包含更长的序列和更丰富的对象属性。



(4) Cityscapes 3D: (<https://www.cityscapes-dataset.com/cityscapes-3d-dataset-released/>)

- 专注于城市场景的三维感知。该数据集提供高分辨率的城市图像，包括详细的语义分割和实例分割注释，以及激光雷达点云数据。Cityscapes 3D数据集致力于促进研究和发展在城市环境中进行3D场景理解的算法和系统。



斯图加特

苏黎世

乌尔姆

蒂宾根

4. 研究数据

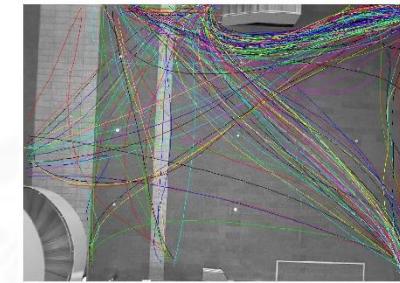
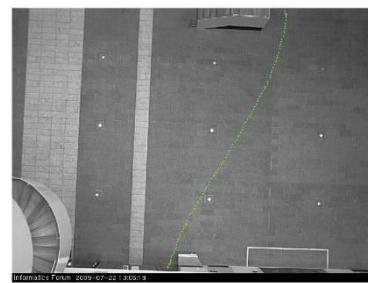


北京交通大学

▲ 相关经典数据集举例：

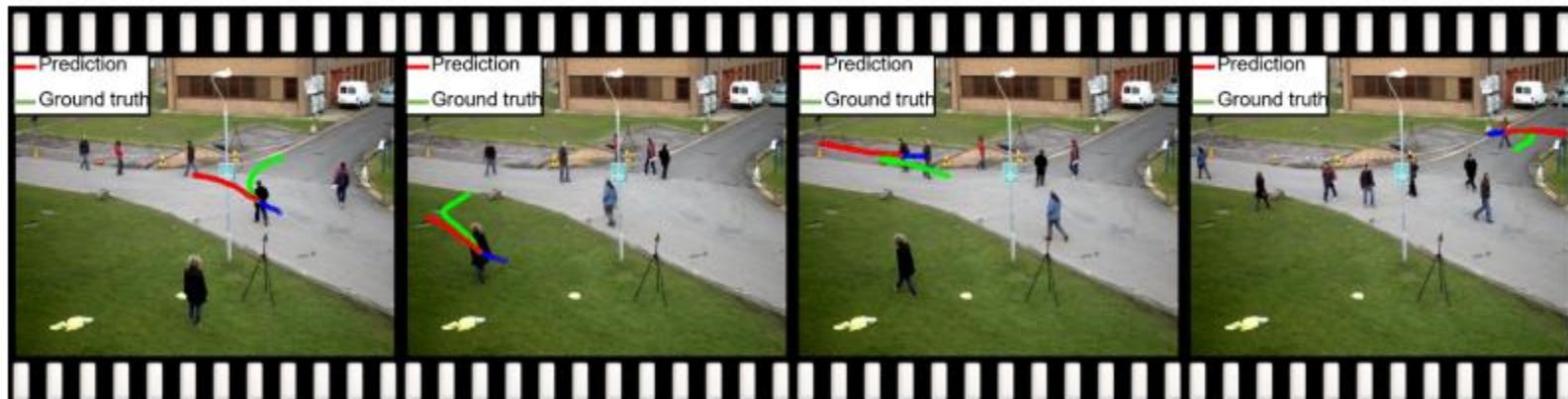
(5) Edinburgh Informatics Forum Pedestrian Database: (<https://homepages.inf.ed.ac.uk/rbf/FORUMTRACKING/>)

- 用于行人检测和追踪研究的数据集。该数据集提供了在不同场景下的高分辨率摄像头图像，以及详细的行人注释，包括边界框和运动轨迹。这为开发和评估行人相关算法提供了实际场景的丰富资源。



(6) ETH/UCY: (<https://www.ee.cuhk.edu.hk/~xgwang/grandcentral.html>)

- 用于行人轨迹分析的研究资源。该数据集提供了在ETH Zurich大学校园内收集的行人运动轨迹数据，包括多个场景和行人相互作用。这为研究人员提供了在复杂环境中进行行人行为分析和轨迹预测的实验基础。



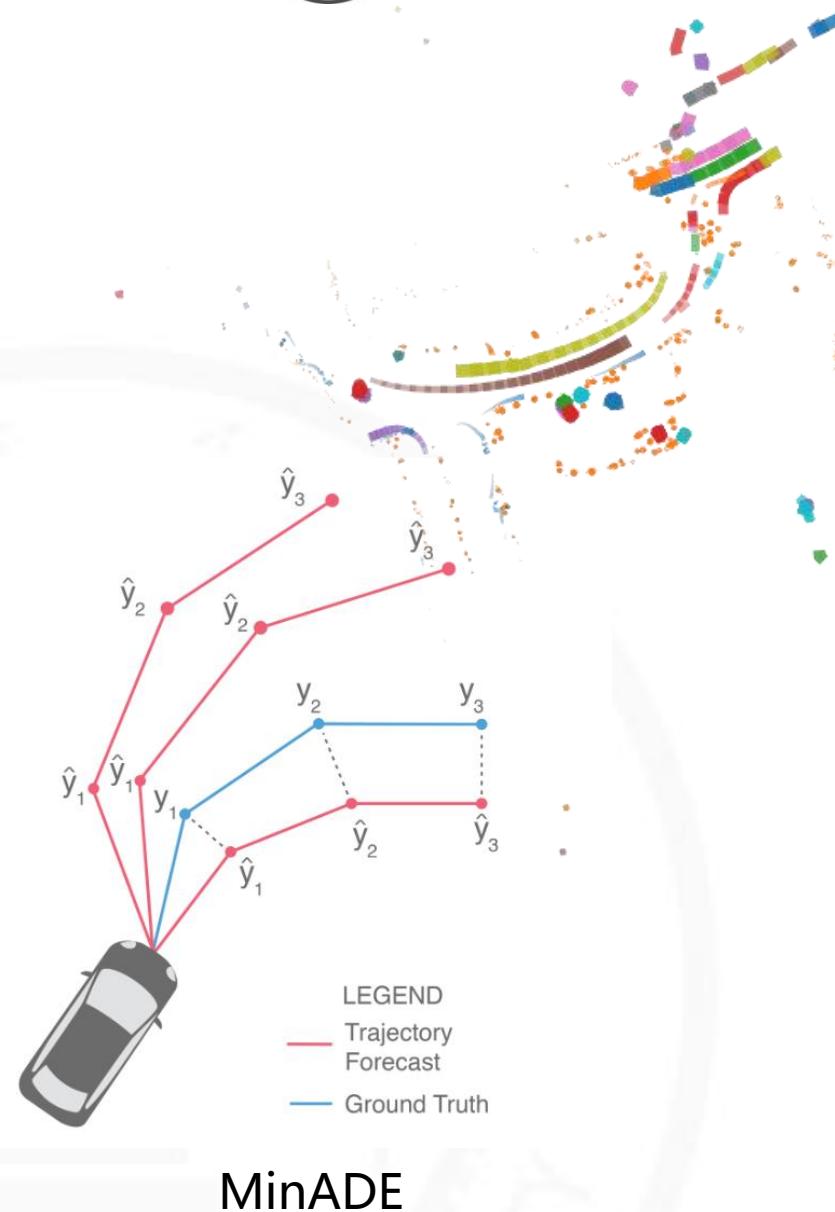
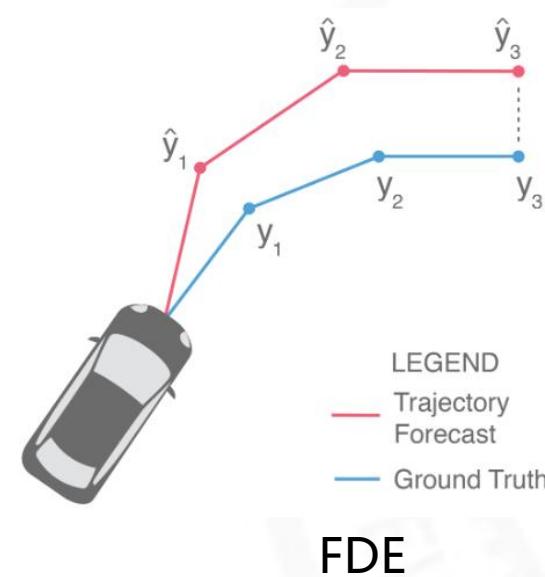
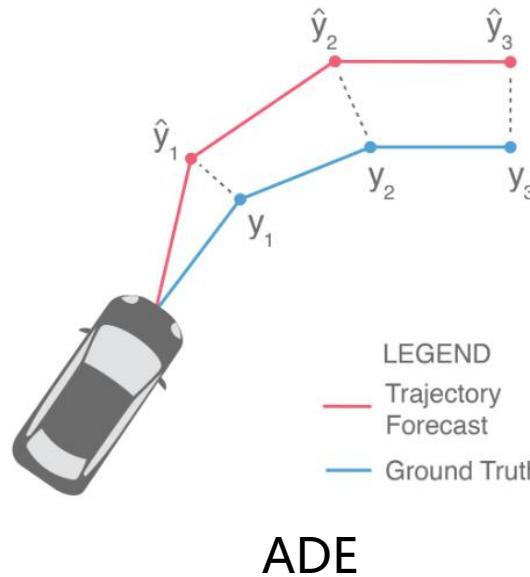
5. 评价指标



北京交通大学

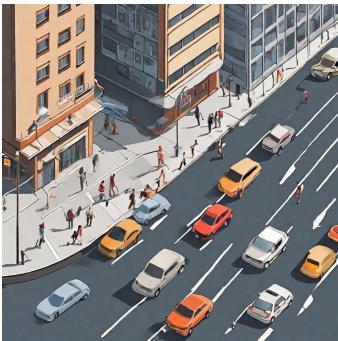
▲ 常用的评价指标: (包括但不限于以下这些~)

- 错过率 (Miss Rate, MR)
- 位移误差 (Displacement Error, DE)
- 最终位移误差 (Final Displacement Error, FDE)
- 平均位移误差 (Average Displacement Error, ADE)
- 最小最终位移误差 (Minimal Final Displacement Error, MinFDE)
- 最小平均位移误差 (Minimum Average Displacement Error, MinADE)
- Brier-最小最终位移误差 (Brier-Minimal Final Displacement Error, b-MinFDE)



6. 应用领域

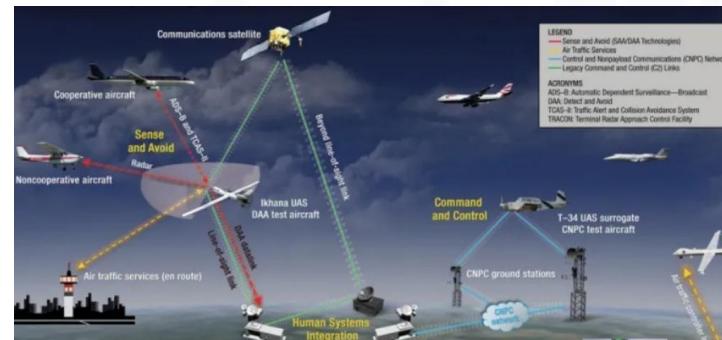
▲自动驾驶领域：预测交通参与者行为、车辆路径规划、防碰撞系统



▲军事国防领域：战场态势感知、目标追踪与打击、运动规划与部队调度



▲航空航天领域：飞行路径规划、空中交通管理、天气预测



6. 应用领域

▲ 物流与供应链管理：货物运输优化、库存管理、交通拥堵预测



▲ 城市规划与交通管理：交通流优化、公共交通规划、城市安全



▲ 体育与运动分析：运动员训练与表现分析、观众体验增强、运动策略规划



北京交通大学

7. 未来展望



北京交通大学

- **多模态数据融合：**
轨迹预测将更关注于多模态数据的融合。通过结合**传感器数据、环境信息、语义信息**等多种数据信息，可以更全面地理解和预测目标的未来轨迹，提高预测的准确性和鲁棒性。
- **跨域自适应学习**
轨迹预测将更聚焦于跨域场景下的自适应学习。通过在**不同环境、不同领域**之间进行**知识迁移和特征共享**，从而实现在新场景中的快速适应和准确预测。
- **时空建模与推理**
轨迹预测将更加注重对目标运动的时空建模和推理。通过考虑目标的**运动模式、环境约束和上下文信息**，可以更精确地预测目标未来的轨迹。
- **可解释性轨迹预测**
轨迹预测将更关注预测结果的可解释性。通过设计**可解释的模型结构和解释方法**，能够提供对轨迹预测结果的解释和理解，增强对预测结果的信任和**可信度**。