

论文综述

自我整理

文献综述：

Motion Forecasting for Autonomous Vehicles: A Survey, <https://arxiv.org/abs/2502.08664>

Trajectory Prediction for Autonomous Driving: Progress, Limitations, and Future Directions, <https://arxiv.org/pdf/2503.03262.pdf>

待看的核心论文：

MTR++

VAD

时序语义分割预测/Occupancy Flow: *Perceive, Predict, and Plan: Safe Motion Planning Through Interpretable Semantic Representations*, *UnO: Unsupervised Occupancy Fields for Perception and Forecasting*(CVPR2024)

快速浏览：文献标题：Learning lane graph representations for motion forecasting，文献标题：
AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting，文献标题：
EqMotion: Equivariant Multi-Agent Motion Prediction with Invariant Interaction Reasoning 类似
SIMPL的研究感觉，文献标题：MotionDiffuser: Controllable Multi-Agent Motion Prediction Using
Diffusion 扩散模型的研究，排名：19文献标题：MTP-GO: Graph-based probabilistic multi-agent
trajectory prediction with neural ODEs，排名：20文献标题：BiTraP: Bi-directional pedestrian
trajectory prediction with multi-modal goal estimation，排名：22文献标题：TrajMAE: Masked
autoencoders for trajectory prediction

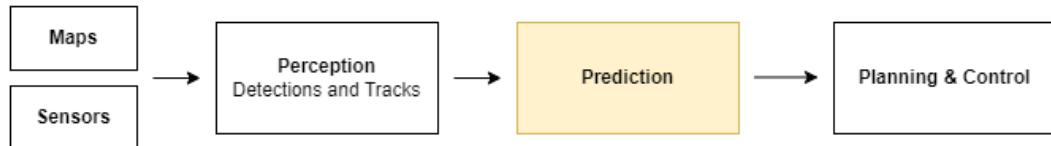
程杰的论文：后续再看；<https://zhuanlan.zhihu.com/p/18319150220>

使用混合高斯轨迹建模：轨迹多模态的最简单表示，K 模式概率，K 轨迹回归，每个轨迹有 T 个航点；模型训练时，只有最接近的mode会受到监督(赢家通吃)，缺点是，概率训练导致模态崩溃(坍塌)和不稳定，代表论文*MultiPath: Multiple Probabilistic Anchor TrajectoryHypotheses for Behavior Prediction*

目前我们的思路还是轨迹预测或者占用预测，我们能否直接预测未来场景(video or Pointcloud)? 这就是world model要做的事情了，进行传感器数据预测。world model代表性论文有GAIA-1: A Generative World Model for Autonomous Driving, DriveGAN: Towards a Controllable High-Quality Neural Simulation, Vista: A Generalizable Driving World Model with High Fidelity and Versatile Controllability(CVPR2024), GenAD: Generative End-to-End Autonomous Driving (CVPR2024)

1.1 模块功能

[轨迹预测](#)在自动驾驶中起到了承上启下的作用，它不仅是环境感知的延伸和决策规划的依据，还是风险评估的基础和人机交互的桥梁。通过准确的轨迹预测，自动驾驶系统可以更好地适应复杂多变的交通环境，提高行驶的安全性和效率。



1.2 研究综述

1.2.1 Physical-based 经典物理推导

运动学模型：基于经典物理方程（如CV恒速、CA恒加速、自行车模型）描述运动规律

核心优势：轻量化、计算快、可解释性极强、无数据依赖

主要缺陷：无法建模人类随机行为（如突发变道）、难以捕捉智能体间复杂交互

典型方案：CV/CA模型，[Constant velocity model](#) (CV) , [Constant Turn Rate and Acceleration](#) (CTRA) 它描述的是一个物体在做匀速转弯和匀加速运动

1.2.2 Machine Learning Based 机器学习类

概率模型类	用统计模型 (GMM、HMM、DBN) 建模轨迹的概率分布与时序依赖	概率输出能力强、小数据场景适配性好、可解释性中等	难以处理高维非线性特征、复杂交互建模能力弱	高斯混合模型 (GMM) 、隐马尔可夫模型 (HMM)
非深度学习模型	基于传统机器学习 (SVM、决策树) 学习轨迹特征与运动模式的映射	训练快、部署成本低、对硬件要求低	特征工程依赖强、复杂场景泛化差	支持向量机 (SVM) 、随机森林

1.2.3 Deep Learning Based 深度学习类

1.2.3.1 研究综述总结

核心两个问题：环境信息编码， 预测轨迹解码

环境信息编码

Encoder 编码问题：栅格化 -> 向量化表征 -> Transformer Model(Traj based --> Goal based --> Query based) --> VAEs/Diffusion(时耗问题)

信噪比问题；细节信息难以捕捉；

- 栅格化渲染：

201910 MultiPath: Multiple Probabilistic Anchor Trajectory Hypotheses for Behavior Prediction

202006 CoverNet: Multimodal Behavior Prediction using Trajectory Sets

- 向量化场景表征：

202006 VectorNet: Encoding HD Maps and Agent Dynamics From Vectorized Representation

202011 LaneGCN: Learning Lane Graph Representations for Motion Forecasting

- 交互建模 (GNN --> Transformer Attention)

202006 VectorNet: Encoding HD Maps and Agent Dynamics From Vectorized Representation

202011 Learning Lane Graph Representations for Motion Forecasting

Wayformer

SceneTransformer

- 坐标系处理

1、Agent Centric：以agent为第一视角viewpoint-invariant(视点不变)；可学习样本多，训练相对容易。缺点是infer速度慢，agent更新地图和其他agent都需要进行转换计算。

2、Ego-Centric或者Scene-centric场景中心（一般选取ego为中心）：典型论文Scene Transformer；

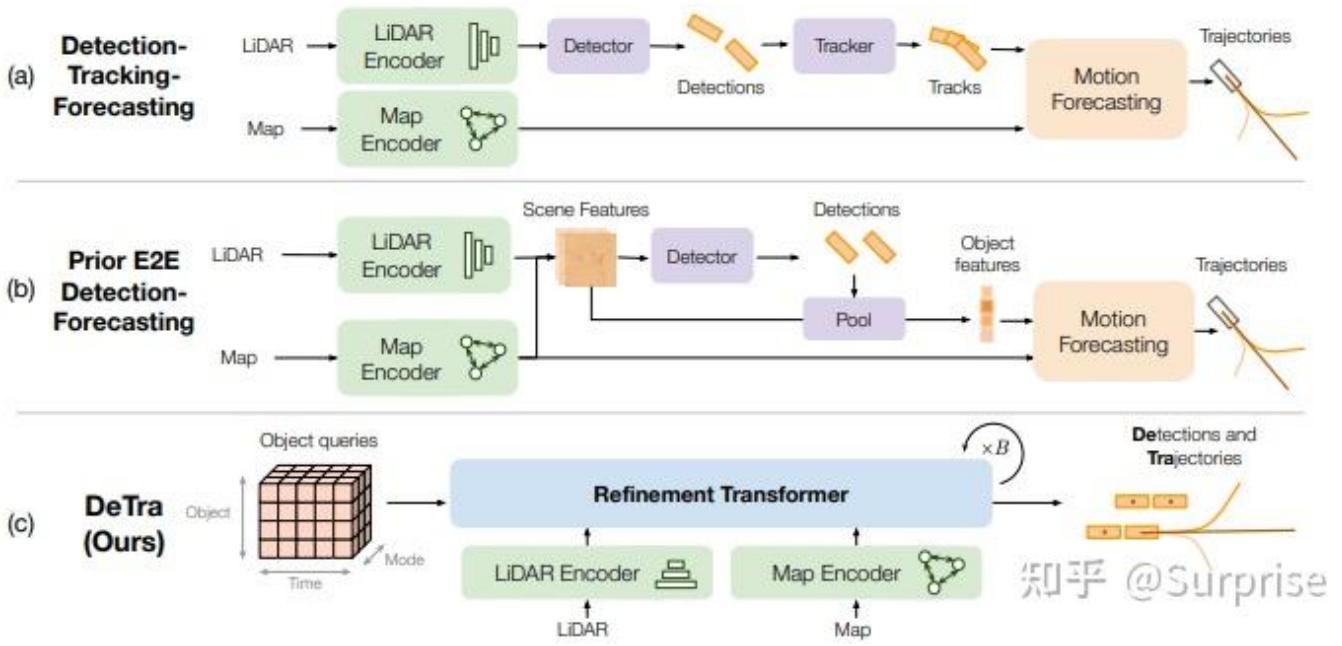
3、Query Centric或Instance Centric：

QCNet

HiVT

- 一段式预测

DeTra: A Unified Model for Object Detection and Trajectory Forecasting



预测轨迹解码

Decoder 预测建模问题：

核心论文串讲：主要解决问题，模型架构，解决思路，贡献

==> 强化学习方法？？没太懂他怎么运作的

- 直接回归：模态坍塌
- 生成式建模：SocialGAN/CAVE，需多次采样构建多模态且无法表达模态概率
- 轨迹Anchor

MultiPath，高纬度分类任务，不利于高准确的预测；

- Goal点Anchor

固定Anchor，依赖Anchor的设计精巧和覆盖程度

- Query
- 边缘预测和联合预测问题

202106 Scene Transformer: A unified multi-task model for behavior prediction and planning

1.2.3.2 研究历史

201910 MultiPath：栅格化场景表征（Rasterized image）> 多模态建模（轨迹锚点+高斯分布模型）

MultiPath: Multiple Probabilistic Anchor Trajectory Hypotheses for Behavior Prediction

Yuning Chai* Benjamin Sapp* Mayank Bansal Dragomir Anguelov

Waymo LLC
{chaiy, bensapp}@waymo.com

历史问题和主要贡献：

- 解决核心问题：==》核心不在环境信息编码，而在于预测轨迹解码

论文聚焦自动驾驶场景中的**多智能体未来轨迹预测问题**，核心目标是解决轨迹预测的“**多模态不确定性**”挑战。传统方法：

- 单模态方法（传统回归）只预测最可能的单条轨迹，无法处理多模态场景；
- 生成式方法需要通过噪声采样来生成轨迹（CVAE/GAN），推理效率低且采样随机性难复现且无法量化不确定性（不能表达模态概率）
- 直接学习高斯混合分布模型，导致**模态坍塌问题**
(<https://chat.deepseek.com/share/7e60vkoan1k56o0viz>, 赢家通吃的梯度更新+走捷径的损失最小化策略+缺乏显式多样性激励)

- 主要贡献：

- 分层不确定性建模：意图不确定性（锚点轨迹选择）+控制不确定性（高斯分布）
- 锚点轨迹：通过无监督学习k-means或均匀采样得到固定锚点，作为多模态基础，避免模态坍塌问题和生成式采样依赖问题，支持模态概率输出；
- 高斯分布：轨迹锚点上的每个轨迹点认为是**独立同分布（潜在问题是导致轨迹的时序约束能力差）**，支持单次前向推理得到所有轨迹点的高斯分布参数（效率远超生成式采样方案）；

环境信息编码：

- 环境信息编码：

- 栅格化渲染：采用鸟瞰图BEV渲染，包含两类环境特征，静态特征（车道线、停止线、限速等）和动态特征（历史智能体状态、历史交通灯信息）。

BEV空间分辨率：输入 BEV 图像为 400×400 像素，对应真实世界 $80m \times 80m$ （即 1 像素 = 0.2m），输入Tensor [B, C, H, W]，其中C是特征通道数。**如Ci是当前时间步的智能体轨迹特征通道，[1, Ci, h, w]为0或1值，表示该位置是否被障碍物边界框覆盖。**

(1) 输入内容与通道构成

论文将“静态环境 + 动态智能体”信息渲染为 **鸟瞰图 (BEV)** 3D 张量，通道数 (C_in) 由两类特征组成：

特征类型	具体内容	通道数
静态环境特征	道路语义 (车道 / 停止线 / 人行横道)	3
	距离道路边缘的距离图	1
	限速标识	1
动态时序特征	过去 5 个时间步的交通灯状态	5
	过去 5 个时间步的智能体轨迹 (定向边界框)	5
总计	-	15

2、场景特征提取：轻量化ResNet卷积处理，输出BEV场景特征图

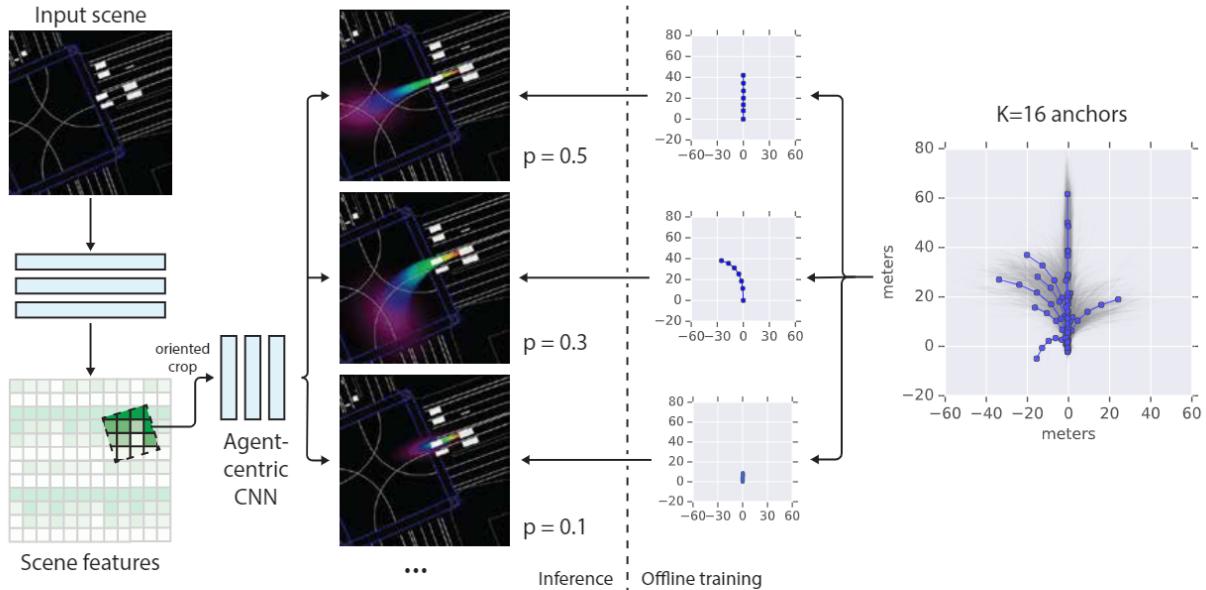


Figure 1: MultiPath estimates the distribution over future trajectories per agent in a scene, as follows: 1) Based on a top-down scene representation, the Scene CNN extracts mid-level features that encode the state of individual agents and their interactions. 2) For each agent in the scene, we crop an agent-centric view of the mid-level feature representation and predict the probabilities over the fixed set of K predefined anchor trajectories. 3) For each anchor, the model regresses offsets from the anchor states and uncertainty distributions for each future time step.

- 优缺点：

优点：有效利用成熟的CNN-based的特征提取方法

缺点：

栅格化渲染表征的缺点：

- 1、空间栅格分辨率不足，难以表征好高精度的信息；
- 2、稀疏的有效信息，栅格化渲染导致图片中存在大量无用信息；
- 3、长距离关系建模困难（多层卷积虽然能缓解感知视野小的问题，但多层处理导致有效梯度信号较小，不如Self-Att机制直接有效）
- 4、丢失连续的物理信息，比如速度，难以渲染表达；

预测轨迹解码：

- 轨迹解码：

1、分层不确定性建模：锚点轨迹+高斯分布：[论文综述](#)

2、Agent-Centric预测：以目标智能体为中心裁剪局部区域，并通过可微分的双线性变化旋转轨迹（消除航向差异），经过四层卷积得到两类参数：锚点概率 + 高斯分布的偏移量和协方差

3、Loss处理：WTA赢者通吃 + 最小化负对数似然函数

4、WTA的策略原因 ([解释轨迹预测 WTA 策略 - 豆包](#), [WTA为何能解模式崩塌 - 豆包](#))：WTA 通过“非此即彼”的硬分配机制，强制模型的不同模态“各自为政”地学习特定行为模式，彻底打破了软分配中‘大锅饭’式的参数混淆局面。

==>模态混淆问题：软匹配会让模型在多个候选轨迹间“平均分配误差”，导致不同运动模式的特征被混淆。需要通过硬分配来强化模态特征区分。

==>模态崩塌问题：模型会倾向于输出“损失最小的单一模式”（如训练集中占比高的“直行”），忽略小众但关键的模式（如“U-turn”），最终多模态覆盖不足。WTA能够：硬分配来强制模态特征分离，以及强化小众模态（模态竞争机制，防止主模态垄断）

==>缺点：1、硬分配方法可能忽略真实轨迹与多个候选轨迹的潜在关联，但在实践中，它是多模态预测的最优折中选择：以小精度损失换来了多模态表征能力的提升；2、对模态初始化敏感，需要良好的锚点或者Query设计；

- 优缺点：

优点：

缺点：

202006 VectorNet：向量化场景表征 + 交互编码（GNN图神经网络）> MLP直出

代表论文：

202006 VectorNet: Encoding HD Maps and Agent Dynamics From Vectorized Representation

VectorNet: Encoding HD Maps and Agent Dynamics from Vectorized Representation

Jiyang Gao^{1*} Chen Sun^{2*} Hang Zhao¹ Yi Shen¹
 Dragomir Anguelov¹ Congcong Li¹ Cordelia Schmid²
¹Waymo LLC ² Google Research

{jiyanggao, hangz, yshen, dragomir, congcongli}@waymo.com, {chensun, cordelias}@google.com

历史问题和主要贡献：

历史问题：栅格化场景表征方法的缺陷

核心贡献：提出了向量化的场景表征方法，并构建分层图神经网络来提取特征信息以及交互关系

环境信息编码：

1、提出向量化的场景表征方法：

$Vi = [dis, die, ai, j]$, 有序的向量表达, ai表达特征属性如障碍物类型、时间timestep、road特征类型、速度限制; j表达Polyline的整数index;

Ego-Centric: 相对位置坐标，使得特征具有相对目标Agent的位置不变性

2、分层图神经网络：

Polyline Subgraph: 多层GCN+max pooling, 实现有序的向量化子序列信息的环形提取，输出[N, D]

Global interaction graph: 自注意力机制实现子图信息之间的交互关系

3、随机mask策略提升网络模型鲁棒性

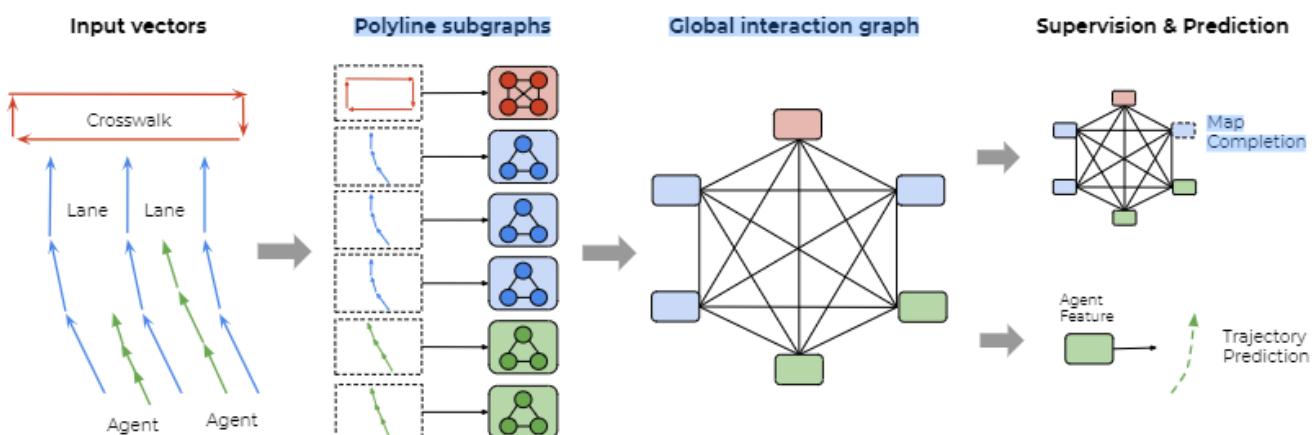


Figure 2. An overview of our proposed VectorNet. Observed agent trajectories and map features are represented as sequence of vectors, and passed to a local graph network to obtain polyline-level features. Such features are then passed to a fully-connected graph to model the higher-order interactions. We compute two types of losses: predicting future trajectories from the node features corresponding to the moving agents and predicting the node features when their features are masked out.

预测轨迹解码：

1、MLP实现 = Linear + LayerNorm + ReLU + Linear, 输出60个轨迹，没有亮点

202008 TNT: VectorNet > 多模态建模 (Goal锚点+MLP轨迹回归)

202008 TNT: Target-driveN Trajectory Prediction

通过“目标驱动”的分阶段框架，实现可解释（目标点显式建模意图）、高效（无测试采样）、高精度（多模态覆盖完整）的轨迹预测，支撑自动驾驶安全决策。

历史问题和主要贡献：

- 历史问题：多模态预测能力

1、生成式方法(CVAE/SocialGAN)：把意图作为隐变量建模，在部署时候需要通过test-time sampling
【测试时采样是一种在模型推理阶段（而非训练阶段）生成多个输出候选，并从中选择最佳结果的技术。】随机多次采样来获得多模态，计算成本大且多模态不可控且可解释性差。

2、单模态方法：直接回归，模态坍塌

3、轨迹Anchor方法：典型MultiPath，轨迹为锚点，锚点选择和偏移回归。问题：高纬度锚点不易做分类；

- 主要贡献

1、Goal Anchor：相比轨迹锚点，维度降低很多，通过均匀采样或者专家知识轻松离散化实现。

2、轨迹预测的多模态问题，分解为意图不确定（目标Goal选择的概率分布问题）+运动控制不确定性（轨迹生成）。Goal锚点显示建模了目标状态，将意图不确定性转换为目标分布的概率问题，而非隐变量黑箱建模。

环境信息编码：

1、复用VectorNet

预测轨迹解码：

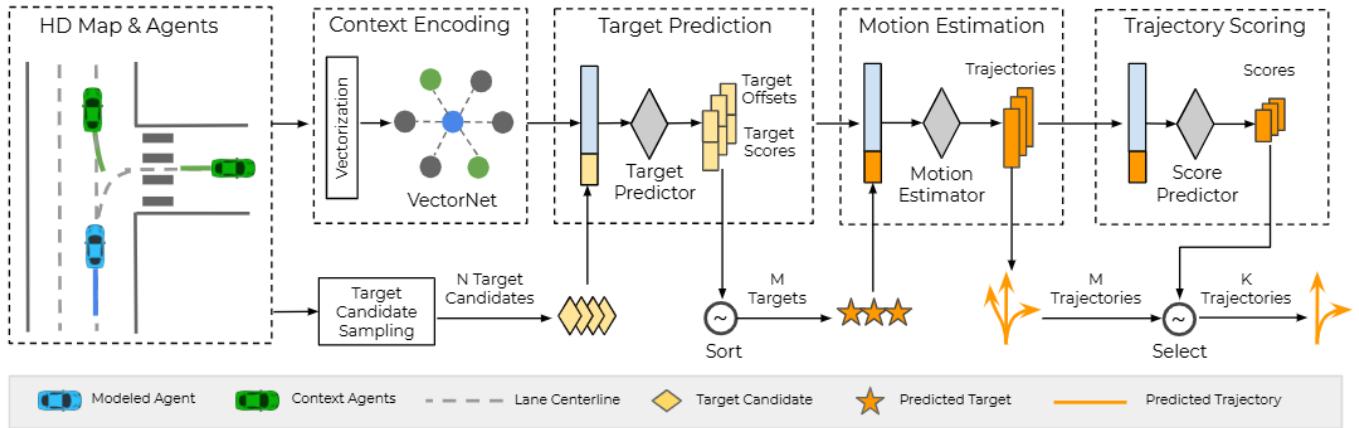


Figure 2: TNT model overview. Scene context is first encoded as the model’s inputs. Then follows the core three stages of TNT: (a) *target prediction* which proposes an initial set of M targets; (b) *target-conditioned motion estimation* which estimates a trajectory for each target; (c) *scoring and selection* which ranks trajectory hypotheses and outputs a final set of K predicted trajectories.

预测轨迹解码要点：

- 1、Agent-centric：坐标归一化，采用智能体中心坐标系，相除绝对位置干扰；
- 2、“Goal预测→Goal condition的轨迹生成→轨迹评分与选择” 三阶段流水线
- 3、Goal预测：
- 4、Goal Condition的轨迹生成：假设给定Goal的轨迹分布是高斯单模态的（轨迹点分布都是独立同分布），采用教师监督（teacher forcing）训练；
- 5、Target Goal越密集效果越好，对于车，差不多在1.0m的间距是最好的；对于行人，差不多是0.5m的间距是最好的；
- 网络结构
 - 1、目标Target预测（2-MLP）：N=1000个候选点，给出每个候选点的($\pi_i, \Delta_x, \Delta_y$)，其中 π_i 就是图2中的Target Scores得分或者概率值；
==> 排序得到Top-M=50的目标target点；
==> 论文对于 Δ_x 和 Δ_y 采用了Huber Loss，其结合了L1 MAE和L2 MSE Loss的优缺点，在误差大的时候用L1抑制异常值的影响，在误差小的时候用L2避免收敛震荡，保证收敛精度；
 - 2、2-MLP实现轨迹预测：训练时用教师监督，HuberLoss回归每一步的轨迹点欧式距离差异；对于每个Target Pos都回归出一条轨迹线；
 - 3、2-MLP实现轨迹评分和选择：输入前序网络的轨迹预测+环境隐特征X，输出轨迹预测集的评分
==> 真值轨迹的得分： $D(s_i, s_j)$ 两条轨迹的最大轨迹点距离，对轨迹集与真实轨迹都计算出D，D序列再用softmax归一化，再计算交叉熵；
 - 4、最终的轨迹选择算法（排除近似重复的轨迹）NMS算法：根据分数排序轨迹，然后顺序处理，处理新轨迹时，确认它是否和其他已选择的轨迹足够远。足够远就选择，否则就排除；
 - 5、损失计算 Loss：三个阶段的损失相加

6、网络的关键参数：

N=1000, M=50, K=6,2-MLP隐层64维度，50个epochs, Adam lr0.001 bs128

遗留问题

- 论文的第二阶段的Condition Goal的运动学预测，有两个关键假设：

第一：未来时间步相互之间是独立分布的（短时间情况下相互时间步之间没有很强的运动学依赖，更多是target目标导向依赖）；21 31 33 34包括MultiPath，都是自回归式/传统序列预测方法的依赖前面已给出的预测结果。

第二：给定目标target的轨迹分布是高斯单模态的（短时间成立但长时间不成立，左右避就不是同一个模态）

==> 因为离散目标target+短时轨迹，可以认为它是单模态的高斯分布。因此可以直接用轨迹回归方式处理；

==> 对于长时间问题，论文提出划分多个短时target，交互迭代。比如先出3s再出5s的轨迹。

==> 论文的验证数据集都是预测3s轨迹的，所以可以满足假设1&2

202106 Scene Transformer: TransformerBased的交互建模&掩码策略应对不同任务 > 联合预测 (一次推理预测所有Agent)

202106 Scene Transformer: A unified multi-task model for behavior prediction and planning

历史问题和主要贡献：

- 历史问题

1、无法一次性预测所有agent的未来轨迹；

2、注意力机制效率低：处理“智能体 - 时间 - 道路图”多轴关系时，要么扁平化多轴（计算复杂度高），要么独立处理单轴（丢失跨轴依赖）；

3、多智能体预测问题：现有方法多为“边际预测”（独立预测每个智能体，无法保证多智能体未来行为一致如轨迹重叠），联合预测要么面临组合爆炸（边际预测组合），要么依赖迭代采样（逐智能体生成，效率低），无法生成一致的联合未来；

- 主要贡献

1. 掩码机制：借鉴BERT的掩码策略，通过不同掩码定义不同任务（行为预测、条件预测、规划），单模型适配多任务，无需修改网络结构；

2. 因子化注意力机制&交叉注意力机制信息融合：一：交替在“时间轴”和“智能体轴”施加自注意力，而非扁平化多轴，平衡效率与跨轴依赖捕捉；二：CrossAttention实现智能体特征与静态/动态道路图特征融合；
3. 场景中心的联合预测（同时也支持边缘预测任务）：直接输出多组一致的联合未来（[F,A,T,7]），避免组合爆炸和迭代采样，确保多智能体行为无冲突；
4. 一次性预测所有障碍物的轨迹；

环境信息编码：

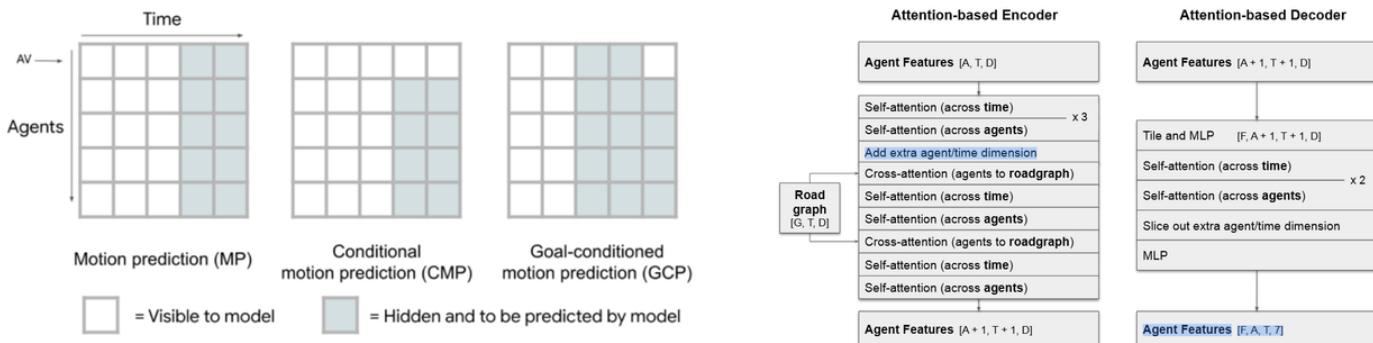


Figure 2: **Single model architecture for multiple motion prediction tasks.** Left: Different masking strategies define distinct tasks. The left column represents current time and the top row represents the agent indicating the autonomous vehicle (AV). A single model can be trained for data associated with motion prediction, conditional motion prediction, and goal-directed prediction, by matching the masking strategy to each prediction task. Right: Attention-based encoder-decoder architecture for joint scene modeling. Architecture employs factored attention along the time and agent axes to exploit the dependencies in the data, and cross-attention to inject side information.

- 数据预处理

1、Ego-Centric或Scene-Centric

2、位置与时间嵌入：用正弦位置编码嵌入空间坐标和时间步，增强模型的位置/时序感知；

Pos Embedding的做法： ==> A和T都做了。T很好理解。A是因为，每个车在ego-centric下的坐标值很大，不好学。所以每个车的输入位置值都是基于各自车当前位置的相对坐标值。然后，pos embedding是每个车当前时刻的位置编码。

3、掩码生成：根据任务生成隐藏掩码 ([A,T])，标记“可见/需预测”的智能体-时间步对。

- 编码过程

1、**因子化注意力机制的编码方式：**场景encode方式是factorized attention，交替在“时间轴”（捕捉智能体自身时序依赖）和“智能体轴”（捕捉智能体间交互）施加自注意力，避免 $O(AT \times AT)$ 复杂度，改为 $O(A \times T^2 + T \times A^2)$ ；

2、**交叉注意力机制信息融合：**Cross Attention实现智能体特征与静态/动态道路图特征融合；

3、理解Add extra agent/time dim的操作。3.3节介绍的内容。类似Bert的做法。

==> 通过两个维度提取/收集agent和time维度的综合特征信息，最终经过2-layer MLP输出每个Future-F维度的场景概率。

==> 感觉还是没有很好理解这一点

==> 为何不放在最后一层？理解：取mean的操作并不能很好代表综合信息，还需要后续多层att操作；

预测轨迹解码：

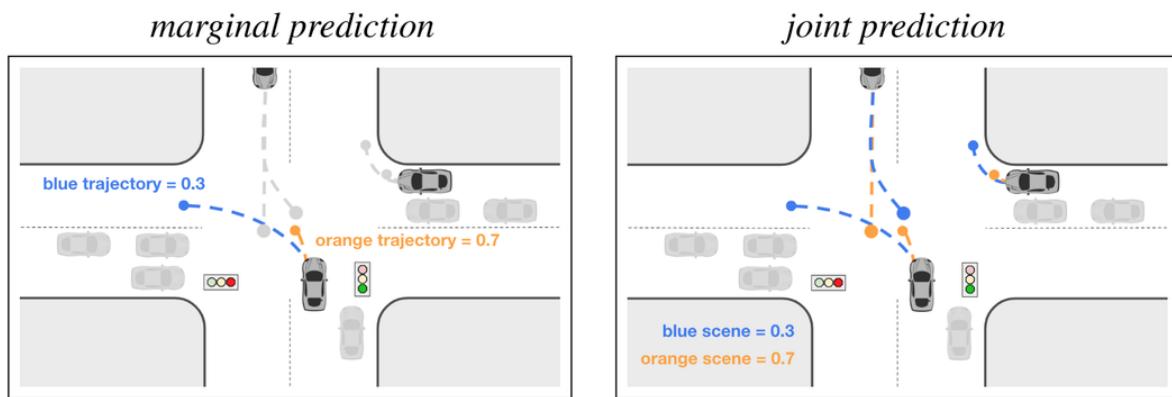


Figure 1: **Joint prediction provides consistent motion prediction.** Illustration of differences between marginal and joint motion prediction. Each color represents a distinct prediction. Left: Marginal prediction for bottom center vehicle. Scores indicate likelihood of trajectory. Note that the prediction is independent of other vehicle trajectories. Right: Joint prediction for three vehicles of interest. Scores indicate likelihood of entire scene consisting of trajectories of all three vehicles.

- 解码过程

1、多未来初始化：复制编码特征 F 次 ($F=6$)，拼接独热编码区分未来，经 MLP 输出[F,A,T,D]；

2、解码器注意力：交替时间 / 智能体自注意力，细化多未来轨迹；

3、预测头：2 层 MLP 输出[F,A,T,7] (3D 位置 + 3 维拉普拉斯不确定性 + 1 维航向)

场景概率头：2层MLP输出[F,A,T,6]，6个场景模态，对应每个联合场景模态里的概率；

4、Loss包含轨迹回归loss以及最优轨迹分类loss

==> 联合预测轨迹回归loss：聚合所有智能体损失，选择最匹配的未来反向传播；“**损失值排序 + 硬选择反向传播**”，也有点赢家通吃的感觉。

==> 最优轨迹分类loss，应该还是使用赢家通吃的策略

- 掩码策略：通过不同掩码适配 3 类任务：

行为预测 (BP)：掩码所有智能体未来时间步；

条件行为预测 (CBP)：掩码部分智能体未来，保留目标智能体完整轨迹；

目标导向规划 (GDP)：掩码 AV 未来轨迹，保留 AV 最终目标位置；

并且，多种conditional任务同时训练，实验结果表明，这并不影响单个任务的表现

- 联合预测和边缘预测是如何实现的？通过损失函数实现。

1、联合损失：聚合所有智能体损失，选择最匹配的未来反向传播；

2、边际损失：每个智能体独立选择最优未来，反向传播。

具体来说：

3.4节介绍，也可以看附录D，Figure 7。

==> 联合预测，基于[F,A] tensor，对于每个F计算所有agent的loss总和，再取最小的loss；

--> 只在F维度下，计算最可能接近真值的网络参数。真值是联合预测的结果，接近真值就是联合预测。

==> 边缘预测，基于[F,A] tensor，先计算每个A的所有future里的最优/最小loss，然后把所有loss累加起来；

论文问题

1、Scene-centric对非Ego Agent是不公平的，模型不能学习到有关场景的视角不变性；

2、时耗长，这是all to all的建模，时间复杂度高；

202111 DenseTNT：信息交互编码（Transformer Att） >

202111 DenseTNT: End-to-end Trajectory Prediction from Dense Goal Sets

202207 Wayformer：信息交互编码（Transformer Att） >

202207 Diffusion简短看一下

202112 MultiPath++（借鉴意义一般）：向量化表征 > 可学习Anchor Embedding

历史问题和主要贡献：

- 解决的核心问题

1、多模态输入信息的处理：输入包含静态（道路拓扑、交通灯）与动态（智能体位置、速度、交互）异质信息

2、多模态轨迹输出能力

- 主要贡献：

1. 取代了rasterized image的encode方式，使用类似与vectornet的做法，融合vector的时候用的是MCG (multi-context gating) 的做法。
2. 还是使用了anchor的想法，但是现在会学习latent embedding (可以理解成原来直接的anchor轨迹的潜在表示形式 ==> 实际实现就是query vector)

环境信息编码：

1、采用了类似VectorNet的矢量化信息编码，**不过整体借鉴意义不大**

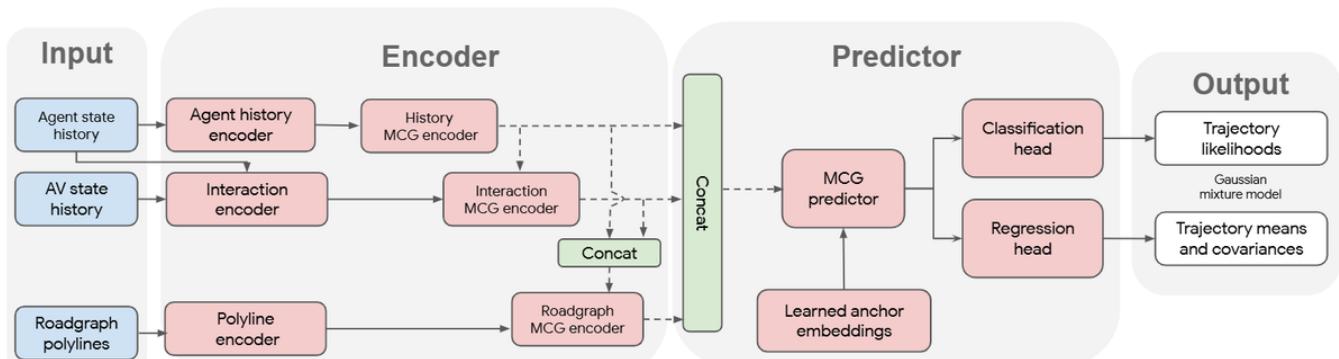


Figure 1: MultiPath++ Model Architecture. MCG denotes Multi-Context Gating, described in Section 3. Blocks in red highlight portions of the model with learned parameters. Dotted inputs to the MCG denotes context features. Each of the encoder MCG outputs aggregated embeddings (one per agent) as shown by dotted arrows. On the other hand, the predictor MCG outputs one embedding per trajectory per agent

预测轨迹解码：

- 1、去掉静态Anchor，采用可学习的Anchor锚点
- 2、仍然采用WTA策略，防止模式坍塌问题；

1.3 其他

[Lime：通俗理解GMM Loss](#)

K个模态就是K个高斯分布的混合模型；

1.4 预测内部现在的挑战（挑战和未来的研究）

模块化功能定义有冲突：不仅仅是横向意图，还包括了纵向意图的功能（除非下游对交互建模非常友好）

一段式预测：

轨迹预测面临的挑战

1. 被遮挡情况下的agent轨迹预测：目标没有被检测出，突然出现时轨迹预测往往不够精准
2. 不同agent之间的交互：车辆间博弈交互，往往会影响各自的行驶轨迹
3. 意图不确定性问题
4. 信噪比问题

1.x 文章的核心观点

<https://zhuanlan.zhihu.com/p/30519437101>

Anchor-Free Decoding（无锚点的解码）

与基于锚点的解码方法不同，无锚点的解码方法不依赖于预定义的锚点，而是直接从解码器输出预测轨迹。这种方法避免了锚点带来的空间先验信息限制，但可能导致模型倾向于学习高频模式，而对低频模式的学习不足，从而在长期预测任务中精度下降。为了解决这一问题，提出了一种可学习锚点的解码范式，结合了基于锚点和无锚点方法的优点。

论文：Trajectory Prediction for Autonomous Driving: Progress, Limitations, and Future Directions

自动驾驶轨迹预测核心方法对比表

方法大类	子类/具体类型	核心逻辑	核心优势	主要局限	代表模型/技术	适用场景
物理基方法	运动学/动力学模型	基于经典物理方程（如CV恒速、CA恒加）	轻量化、计算快(<10ms)、可	无法建模人类随机行为（如突发变道）、难以捕	CV/CA模型、卡尔曼滤波、蒙特卡洛方法	简单低速场景（如小区道）

	解释性极强、无数据依赖	捕捉智能体间复杂交互		路) 、应急兜底预测	
	滤波类方法	通过概率滤波(卡尔曼滤波、粒子滤波)融合观测数据,修正轨迹预测结果	抗传感器噪声能力强、实时性好	对非线性场景适配差、长时预测误差累积	扩展卡尔曼滤波(EKF)、粒子滤波(PF)
机器学习基方法	概率模型类	用统计模型(GMM、HMM、DBN)建模轨迹的概率分布与时序依赖	概率输出能力强、小数据场景适配性好、可解释性中等	难以处理高维非线性特征、复杂交互建模能力弱	高斯混合模型(GMM)、隐马尔可夫模型(HMM)
	非深度学习模型	基于传统机器学习(SVM、决策树)学习轨迹特征与运动模式的映射	训练快、部署成本低、对硬件要求低	特征工程依赖强、复杂场景泛化差	支持向量机(SVM)、随机森林
深度学习基方法	RNN/LSTM类	利用循环神经网络捕捉轨迹的时序依赖关系,建模连续运动模式	时序特征提取能力强、模型结构简单、易训练	长时依赖捕捉不足(梯度消失)、空间特征建模弱	Social LSTM、Trajectron++、ST-LSTM
	CNN类	用卷积操作提取轨迹的空间特征(如相对位置、区域分布),适配网格/图像输入	空间特征提取高效、并行计算能力强、推理速度较快	时序长依赖捕捉弱、难以显式建模交互关系	CoverNet、Social-STGCNN、CNN-LSTM混合
	GNN类	将智能体/地图抽象为图节点,通过图卷积/注意力建模显式交互(智能体-智能体、智能体-地图)	交互关系建模直观、支持动态拓扑结构、适配路网拓扑	图构建依赖先验知识、长程交互捕捉有限	LaneGCN、EvolveGraph、VectorNet
	Transformer类	基于自注意力/交叉注意力机制,全局捕捉时空依赖与多源特征关联	长程交互建模能力强、多源特征融合效果好、泛化性优	参数量大、计算成本高、推理速度较慢	AgentFormer、HiVT、Scene Transformer

	自编码器/VAE类	通过编码器-解码器结构学习轨迹的低维潜表示，支持多模态生成与不确定性建模	多模态轨迹覆盖全、可量化预测不确定性、抗噪声能力强	训练难度高、部分模型可解释性弱	Social-VAE、BiTraP、VAE-GAN混合	需输出多模态候选的规划适配场景
	GAN类	通过生成器-判别器对抗训练，生成真实感强的多样化轨迹	轨迹真实度高、多样性好、适配罕见行为生成	训练不稳定（模式坍塌）、概率建模能力弱	Social GAN、SoPhie、Traj-GAN	需多样化轨迹输出的场景、罕见行为预测
	端到端联合类	检测-跟踪-预测一体化训练，减少模块间误差传播，统一优化多任务	端到端部署便捷、误差传播少、场景适配性强	调试难度高、故障定位难、需大量标注数据	DETR-Predict、Query-based联合模型	高等级自动驾驶（L3+）、全栈集成场景
强化学习基方法	模仿学习类	通过逆强化学习(IRL) /生成对抗模仿学习(GAIL) 学习专家驾驶行为的奖励函数	决策导向性强、轨迹符合人类驾驶习惯、适配复杂决策场景	专家数据需求大、训练不稳定、奖励函数设计依赖经验	IRL-based预测模型、GAIL-Traj	需结合决策的轨迹预测（如换道/超车决策）
	强化学习类	通过环境交互与奖励反馈，学习最优运动模式，优化轨迹的安全性/效率	可主动适应动态环境、轨迹优化能力强、无需大量标注数据	训练周期长、实际场景迁移难、安全性验证复杂	RL-based Planning-Prediction混合模型	动态环境场景、需自主优化的轨迹预测

注：表格严格基于综述论文《Trajectory Prediction for Autonomous Driving: Progress, Limitations, and Future Directions》的分类体系与核心内容整理，覆盖方法的核心特性与适用边界，可直接用于技术选型参考。