基于场景上下文信息的长时多车轨迹预测

杨秋宇[1]，缐凯[2]，郭继孚[2]，朱重远[2]，焦建彬[3†]

（1 中国科学院自动化研究所，北京 100083；2 北京交通发展研究院，北京 100073；

3 中国科学院大学 电子电气与通信工程学院 北京 101408）

（2024 年 1 月 29 日收稿；2024 年 6 月25日收修改稿）

杨秋宇，缐凯，郭继孚，等. 基于场景上下文信息的长时多车轨迹预测[J]. 中国科学院大学学报，DOI:10.7523/j.ucas. 2024. 066.

摘　要　精准地感知周围车辆的未来行动对自动驾驶的安全性保障有着至关重要的作用，本文主要关注于多车长时间轨迹预测这一复杂问题。已有的轨迹预测方法可以大致分为联合预测和边际预测两类，尽管联合预测方法在多车问题上能取得更好的场景一致性，但这两类方法都在长时间预测上存在局限，因为它们无法模拟驾驶员决策随时间的变化。在本文中，我们提出了全新的联合预测方法（TPP），方法的核心是其中的轨迹后处理模块。该模块利用注意力机制构建不同车辆间的交互，通过单车多编码的设计模拟车辆行驶中决策的更新，最终利用解码器生成具有场景一致性的多车轨迹。我们分别在短时预测数据集和长时预测数据集上对方法进行了评估，并与主流的轨迹预测方法进行了对比。结果表明，TPP 方法取得了更好的性能表现。

关键词　车辆轨迹预测，长时间联合预测，轨迹后处理，深度学习，注意力机制

中图分类号：TP399　　**文献标志码：**A　　**DOI:**10. 7523/j.ucas. 2024. 066

Long-term multi-vehicle trajectory prediction with scene contextual information

YANG Qiuyu[1]，XIAN Kai[2]，GUO Jifu[2]，ZHU Zhongyuan[2]，JIAO Jianbin[3]

(1 *Institute of Automation, Chinese Academy of Sciences, Beijing* 100083, *China;* 2 *Beijing Transport Institute, Beijing* 100073, *China*; 3 *School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing* 101408, *China*)

Abstract Precisely perceiving the future actions of surrounding traffic agents is critical for ensuring the safety of autonomous vehicle. This paper mainly focuses on the complicated problem of long-term multi-agent trajectory prediction. Existing trajectory prediction methods can be categorized into joint prediction and marginal prediction. Although joint prediction methods achieve better scene consistency in multi-agent scenarios, both of them fail to reach satisfactory results in long-term prediction tasks due to their inability to capture changes in driver behavior over time. In this paper, we introduce a novel joint prediction method called Trajectory Prediction through Post-processing (TPP). The core of this method is the trajectory post-

† 通信作者，E-mail：jiaojb@ucas.ac.cn

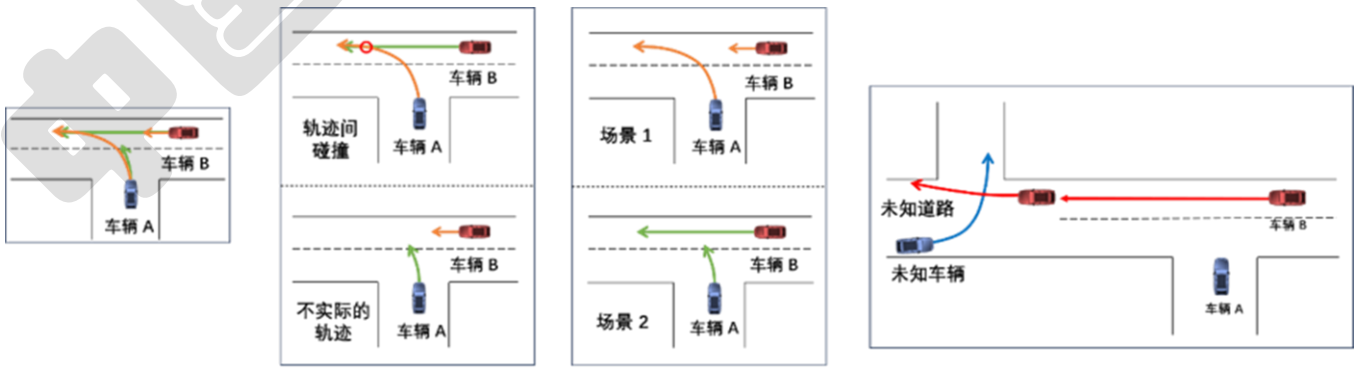
processing module, which utilizes attention mechanisms to model interactions among different vehicles. By representing a single vehicle with multiple embeddings, the module also simulates the changes in behavior during driving. With the help of post-processing module, our method is able to generate consistent multi-agent trajectories. We evaluate TPP on a short-term prediction dataset and a long-term prediction dataset separately, comparing it with mainstream trajectory prediction methods. The results indicate that TPP achieves better performance.

Keywords vehicle trajectory prediction, long-term joint prediction, trajectory post-processing, deep learning, attention-mechanism

在自动驾驶领域中，轨迹预测技术能够极大地提高系统的安全性[1]。利用轨迹预测方法对周边车辆的意图进行感知，能有效地帮助自动驾驶车辆规划出更为安全的行驶路线[2]。本文研究的核心问题是多车长时间轨迹预测任务，该任务的难点主要体现在 2 个方面。首先，提出的方法需要保证场景中所有车辆的预测轨迹具有一致性，即轨迹之间不会发生碰撞；其次，要求模型在长时间的预测任务中仍然能够保持较高的准确度，即预测的轨迹不会偏离车道。现有的轨迹预测方法一般都关注于短时间单车预测问题[3-13]，这类方法也被称作边际预测方法。如图 1(a)所示，模型会独立地为车辆 A 和车辆 B 分别生成一黄一绿两条预测轨迹，预测多条轨迹的目的是覆盖目标车辆所有可能的行驶方向，从而模拟人类驾驶员的主观随机性，这种预测方式在轨迹预测领域被称为多模态预测。

然而将边际预测模型直接应用于多车预测任务会带来一些问题。在边际预测方法中，不同车辆对未来行为的决策彼此独立，这导致生成的未来轨迹间缺乏足够的交互，从而难以保证多车轨迹间的一致性。如图 1(b)所示，以两辆车的预测场景为例，红色车辆和蓝色车辆各自获得了一组多模态轨迹，但由于双方都无法得知对方会采取怎样的驾驶策略，因此最终很可能会产生不合理的全局场景。例如，两辆车可能都选择激进的策略而发生碰撞，也可能都选择礼让的策略而产生不符合现实的预测。

为了解决上述问题，许多研究提出了联合预测方法。这些方法通常是在 1 个边际预测模型的基础上增加了车辆决策的交互机制，以确保不同车辆的预测轨迹之间具有一致性，如图 1(c)所示。目前的联合预测方法主要可以分为两类：基于条件预测的方法和基于模



(a) 多模态预测　(c) 多模态边际预测　(b) 多模态联合预测　(d) 长时间轨迹预测

图 1 轨迹预测任务示意图

Fig.1 Schematic of trajectory prediction task

态选择的方法。基于条件预测的方法[14-17]首先会预测不同车辆的优先通行关系。对于高优先级的车辆，这些方法会直接利用预训练的边际预测模型生成未来轨迹；对于低优先级的车辆，方法中会以高优先级车辆的隐变量作为解码器的条件输入，以此构建高优先级车辆对它的影响。而基于模态选择的方法会假设不同车辆的多模态边际预测之间存在一种无冲突的组合方式。这类方法通过深度神经网络对车辆的不同模态进行评分，为每辆车选取不会与其他车辆发生冲突的模态，从而确保整个预测场景的一致性。

尽管联合预测方法在多车预测任务中取得了不错的效果，但它们仍然无法很好地适用于长时间的预测任务。如图 1(d)所示，在更长时间的预测中，目标车辆的周边环境可能会发生很大的变化，例如附近出现新车辆，或者目标车辆驶入了新道路。这些环境变化导致目标车辆的决策需要实时进行调整，但目前的联合预测方法并不能满足这样的需求。它们只能够根据历史信息做出仅适用于当前时刻的决策，而基于这种决策生成的长时间预测轨迹不可避免地更容易偏离车道以及互相碰撞。

为了提升模型在长时间预测任务中的性能，本文构建了一种基于场景上下文信息的联合预测方法(trajectory prediction through post-processing, TPP)。与传统的联合预测方法相比，TPP 采用了轨迹后处理这一全新的技术路线，充分地利用了目标车辆周边的场景信息，以确保模型能够在多车和长时间这两种预测任务上都取得较好的性能。首先，TPP 使用 1 个预训练的模型为场景中的每辆车生成多模态的边际预测，对车辆的决策进行初步的判断。接着，TPP 通过轨迹后处理模块对生成的轨迹做进一步修正。后处理模块可以分为场景元素交互、时序信息融合和轨迹解码 3 个部分。场景元素交互部分的作用是构建不同时刻多车行为决策之间的交互，以及地图信息对目标车辆的影响。在每个预测时间步上，这部分网络会根据初步预测的未来轨迹，筛选出该时刻目标车辆附近的环境信息，包括周边车辆的未来轨迹和周围车道的位置。接着利用注意力机制计算这些因素对车辆策略的影响，以得到目标车辆在该时刻的特征编码。每辆车在不同时间步上的编码代表了车辆基于不断变化的场景信息做出的不同决策，这种多编码的设计体现了车辆行驶策略的更新。随后，时序信息融合部分会收集同一车辆的所有编码，通过注意力机制构建联系，保证不同时刻上车辆行为决策的连贯性。最后，解码器将不同时刻的编码转换为目标车辆在不同时刻的位置坐标，并组合成为一条完整的预测轨迹。

对于我们的工作，主要的贡献可以总结为如下三点：首先，我们首次提出了基于注意力机制的轨迹后处理方法，实现了多车任务中车辆策略之间的交互。这种后处理方法与已有的条件预测方法和模态选择方法有着显著的区别，代表了一种全新的联合预测方式。其次，我们的方法通过单车多编码的设计实现了车辆决策随时间的更新，在长时间的预测问题中取得了良好的效果，并有效减少了车辆碰撞和偏离道路的情况。最后，我们分别在长时间预测和短时间预测 2 个任务上进行了实验，证明了我们的方法不仅在长时间预测任务中相对于基线算法取得了显著的性能提升，而且仍然能够有效地应用到短时间预测任务中。

1. 车辆轨迹预测算法

在这一节中，我们将对轨迹预测方法进行详细介绍。目前，主流的轨迹预测模型主要可以视作由 3 个部分组成：对周边环境信息的编码，不同车辆之间的交互建模，以及预测轨迹的生成。接下来，我们将对这 3

个部分依次进行介绍。

1.1 环境信息编码

轨迹预测算法首先需要感知目标车辆的状态以及周围的环境。模型进行预测所需的输入信息通常包括目标车辆及其周边车辆的历史轨迹，以及附近的地图信息。对于这些信息，主流的表征方式可以分为栅格化和向量化两种。栅格化方法会将上述的输入信息以图片或图片序列的形式输入，并使用基于卷积神经网络的模型对输入进行编码。文献[3,15,18-19]中地图信息会直接以图片的形式输入，文献[20-21]中车辆历史轨迹信息也同样被转换为图片或矩阵序列的输入形式。而向量化方法[6,12,22]利用了场景中各个元素的方向性特征，将历史轨迹以及车道走向等信息转换为了向量集的形式。这种稀疏编码的表示方式更加简洁高效，并且具有更强的可解释性，从而能够更好地反映数据的内在特征。文献[22]使用图神经网络对向量集内的交互和向量集间的交互进行建模，该方法的网络结构在许多其他模型中也得到了广泛应用[4,7-8,14,16]。文献[5]将车道分为多个首尾相接的车道段，然后使用多尺度卷积网络学习车道图中的节点表征，在文献[9-11]中也使用了类似的车道表征方式。由于我们的方法主要关注于多车的预测问题，采用向量的表征形式更有利于模型捕捉场景中的结构化信息，从而更好地建模车辆间的交互情况。

1.2 车辆交互的建模

按照模型中不同车辆之间交互的程度，轨迹预测可以分为边际预测和联合预测两类。在边际预测方法中，模型主要对车辆的隐变量进行处理，通过更新这些隐变量实现不同个体之间的交互，这个过程中使用的较为广泛的网络结构包括图神经网络[5,12,15,23-25]和基于注意力机制的网络[6,26-31]。而联合预测方法的特点在于模型能够建模车辆在未来时刻的交互，目前的联合预测方法主要可以分为两类。首先是基于条件预测的方法[14]，这类方法会预训练1个神经网络，用于判断场景中不同车辆间的优先级关系。接着，它们先对高优先级的车辆进行边际预测，再以高优先级车辆的轨迹或编码作为低优先级车辆预测的条件输入。文献[15-17]在文献[14]的方法上进行了延伸，文献[15]通过有向无环图构建了多辆车的链式影响关系，实现了场景中所有车辆的联合预测；文献[16]中改为在预测车辆目标点的过程中进行条件预测，而不是在生成轨迹的时候；文献[17]则根据影响时长，选取低优先级车辆中影响力最大的1个模态进行条件预测，从而降低了多模态问题下条件预测的复杂度。另一类是基于模态协调的方法[23,27,32-33]，这类方法以多模态的边际预测模型为基础，利用深度神经网络计算不同车辆不同模态之间的匹配关系，并为每辆车的每个模态输出对应的评分，最终每辆车都会通过评分选择恰当的模态，实现不同车辆之间的联合轨迹预测。这类模态协调的方法通常需要假设在整个场景的预测中，存在一个或多个不会出现冲突的车辆模态组合。但以上两种联合预测方法都无法对长时间预测任务中目标车辆周边环境的变化做出合理的应对。因此，我们的方法选择通过轨迹后处理的方式，在实现联合预测的同时，提升模型在长时间预测任务上的性能。

1.3 生成轨迹

轨迹预测模型中的解码器负责将车辆的特征向量转换为相应的预测轨迹，并可以按照是否基于目标点生成轨迹分成两类。文献[4,7-10,16,34]都是典型的基于目标的预测模型，它们会首先预测车辆在几秒后所处的位置，再根据这个位置和特征向量将中间的轨迹补全。文献[7-8]会沿着车辆所处的道路搜寻，以道

路上的位置作为车辆目标位置的候选；文献[4,9-10]则会计算出车辆最终位置的热力图分布，再利用启发式的算法从分布中采样可能性较高的目标点作为候选。也有许多的轨迹预测方法没有使用基于目标位置的方法。文献[3,26,35-39]考虑到轨迹信息为时序序列的特点，使用了循环神经网络和基于注意力机制的神经网络作为解码器的结构，通过自回归的方式得到车辆位置或车辆加速度的序列。文献[17,40]利用扩散式模型进行轨迹生成，文献[41]通过在解码器部分引入礼貌值的概念实现车辆驾驶风格的控制。文献[5-6,27-28]使用了相对简单的多层感知机（Multi-Layer Preception,MLP）网络，解码器直接以车辆的隐变量作为输入，输出向量的中包含了车辆每个未来时刻的预测信息。基于目标点的轨迹生成方法在数据预处理阶段操作较为复杂，这导致模型在处理新数据时需要花费大量时间。而在大部分的方法中，使用 MLP 作为解码器仍然能够取得较好的效果。鉴于更为复杂的网络结构会带来更高的时间复杂度，TPP 的解码器也选用了 MLP 的结构。

2. 方法

本章中会对提出的 TPP 方法进行详细的介绍，图 2 展示了该方法的整体结构。首先，根据环境中的历史轨迹信息以及道路信息，使用预训练的骨架网络进行初步的边际预测，并通过启发式的方法，利用预测轨迹推断出未来时刻可能存在交互的车辆。接着，使用后处理模块，根据上一步计算出的车辆间关系和地图道路信息，对原始的预测轨迹进行修正与改进，从而实现车辆决策的交互与更新。本章首先会在 2.1 节介绍后处理模块前进行的准备工作，包括场景信息的向量化表示、骨架网络的介绍以及车辆间交互关系的计算方式，接着在 3.2 节详细介绍后处理模块的设计思路以及网络结构，最后在 3.3 节介绍整个方法的训练过程。

2.1 前期准备

场景信息的向量化表示

首先，为了避免生成的轨迹与其他车辆发生碰撞，模型需要知晓目标车辆以及周围车辆的运动态势；其次，为了防止预测轨迹偏离车道，模型还需要感知车辆周围的环境情况。从以上两点出发，TPP 方法的原始输入信息主要包括了车辆的历史轨迹和场景的车道信息，它们的数据格式可以用如下公式表示：

$$\boldsymbol{X_T} = \{\boldsymbol{x}^{i}_{1:t}, : i \in \{1,2,\dots,N\}\}, \quad (1)$$

$$\boldsymbol{X_M} = \{(l_\xi, \boldsymbol{a_\xi}), \xi \in \{1,2,\dots,M\}\}, \quad (2)$$

其中，$\boldsymbol{X_T}$和$\boldsymbol{X_M}$分别代表了车辆的历史轨迹以及场景中的道路信息。N是场景中车辆的数量，M是场景中车道片段的数量，$\boldsymbol{l_\xi}$和$\boldsymbol{a_\xi}$分别是车道片段ξ的位置和语义信

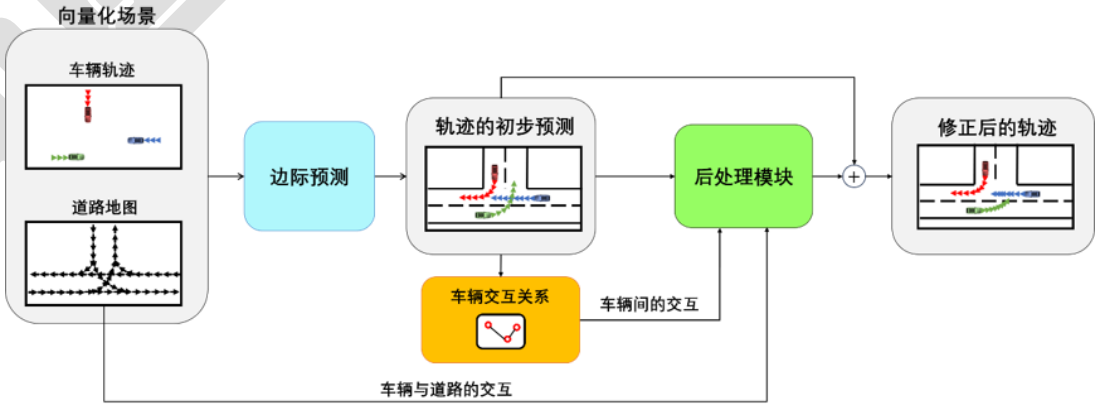


图 2 TPP 方法的整体结构

Fig.2 Framework of the TPP method

息。$\boldsymbol{l_\xi}$主要包含了车道片段中心线的结束点$\boldsymbol{l_\xi^1}$和$\boldsymbol{l_\xi^0}$，而$\boldsymbol{a_\xi}$则包含了相应车道上的一些交通规则，例如车道的限速、限制只能左转等等。在此需要注意的是，一个车道片段只代表了一条车道中的一部分。通过这种分割的方式，便能够用多个距离较短的直线车道对一条距离较长的弧线车道进行表示。由于向量化的表示形式能较好地保留原始输入的结构信息，因此这些信息都会使用向量表示，车辆i的历史轨迹会被表示为：

$$\boldsymbol{V_T^i} = \left\{\boldsymbol{x_i^t} - \boldsymbol{x_i^{t-1}}\right\}_{t=1}^{T}. \quad (3)$$

车道ξ会用结束位置减去开始位置作为向量形式的表示：

$$\boldsymbol{V_M^\xi} = \left\{\left(\boldsymbol{l_\xi^1} - \boldsymbol{l_\xi^0}, \boldsymbol{a_\xi}\right), \xi \in \{1,2,\dots,M\}\right\}. \quad (4)$$

由于向量化处理后会丢失绝对坐标的信息，场景中不同元素之间的相对位置关系也会以向量的形式记录，作为模型的输入，例如车辆j在时间步t中与车辆i的相对位置为$\boldsymbol{x_t^i} - \boldsymbol{x_j^t}$。

预训练的骨架网络

TPP 首先会使用预训练的边际预测模型作为方法的骨架网络进行初步的预测，再在这个基础之上对生成的轨迹进行后续的处理。目前已经有了许多的边际预测方法，综合从性能以及推理速度两方面考虑，TPP 最终选择了文献[8]提出的分层向量变换器(hierarchical vector transformer, HiVT)模型作为骨架网络。HiVT 通过向量的形式表征了车辆轨迹、道路走向等场景信息，并使用以注意力机制为核心的网络结构对这些信息进行综合，最后通过 MLP 解码器生成单个目标车辆的多模态轨迹预测。这样的多模态预测在单车问题中十分常见，例如在包含了N辆机动车场景下，HiVT 模型会为每辆车独立地输出K个模态，分别得到这N辆车的边际预测。但在多车预测的问题下，如果直接将这N个K模态轨迹进行暴力组合，并从中挑选出效果最好的组合作为多车预测的结果，那么方法将会有K^N的指数级别的复杂度。

为此，TPP 对基于 HiVT 的骨架网络进行了两项改进。首先，TPP 对骨架网络的解码器部分进行了重新设计。在解码器中，骨架网络首先会通过 MLP 将目标车辆的特征向量分化为多个编码，再根据这些编码生成对应的多模态轨迹。其中，每个编码都对应着一种车辆模态。相比于原先只通过单个特征向量预测多个模态的方式，提前将信息特征进行分离的做法极大地降低了不同模态间的相似性。其次，TPP 改变了骨架网络的输出逻辑，骨架网络会将场景中的所有N辆车看作一个整体，并为这个整体场景生成K模态的预测。通过这种方式，我们巧妙地将多车模态组合的复杂度从K^N降低为了K，从而减少了模型训练和评估所需要的时间。在具体实现中，这部分的改动主要体现在损失函数的计算上，相应的内容会在 3.3 节中进行介绍。

计算车辆间的交互关系

使用后处理模块对骨架网络生成的预测轨迹做进一步处理之前，首先需要根据这些轨迹确定场景中哪些车辆在未来时刻有着潜在的交互关系。参照文献[14]，TPP 使用了启发式的方法进行判断。对于每一对车辆(i,j)的未来轨迹$\left\{\boldsymbol{y_i^{1:T}}, \boldsymbol{y_j^{1:T}}\right\}$，计算它们轨迹在时空上的最短距离：

$$d = \min_{t_1,t_2} \left\| \boldsymbol{y_i^{t_1}} - \boldsymbol{y_j^{t_2}} \right\|. \quad (5)$$

如果距离d小于某个阈值，说明两辆车的轨迹有可能存在重叠，便认定两车存在交互关系。即使两辆车在初始时刻相距较远，这样的计算方式也能够有效地识别出它们在未来时刻的交互，避免了目标车辆周围突然出现未知车辆的情况。接着，会判断两车到达这个最近距离的先后顺序，即计算如下 2 个时间：

$$t_1, t_2 = \underset{t_1,t_2}{\operatorname{argmin}} \left\| \boldsymbol{y}_i^{t_1} - \boldsymbol{y}_j^{t2} \right\|. \quad (6)$$

因为时间更小的车辆会更先到达交互点，因此相对应地也具有优先通行权。最终构建的交互关系可以看作 1 个有向图，图的节点代表了不同的车辆，边代表了车辆间存在着交互关系，边的方向表示了两辆车之间的通行优先级。

2.2 后处理模块

TPP 的轨迹后处理模块主要可以分为场景元素交互、时序信息融合以及轨迹解码 3 个部分。如图 3 所示，场景元素交互部分指的是图中的场景编码器，它包含了车辆-车辆编码以及车辆-车道编码两部分，它们分别建模了特定时刻上不同车辆策略之间的交互，以及道路信息对车辆决策的影响，保证模型能够很好地适应存在大量交互的多车预测任务。该部分的输出是每辆车在不同时间步上的编码，这些编码也代表了车辆不断更新的行为决策。

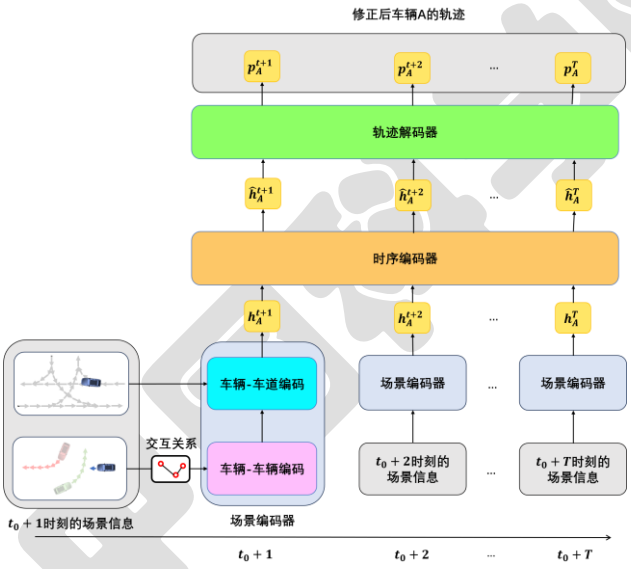


图 3 轨迹后处理模块

Fig.3 Structure of trajectory post-processing module

轨迹后处理模块接着会使用时序编码器构建这些车辆决策之间的联系，进而保证模型生成的轨迹是平滑的序列。通过为每辆车赋予多个编码，TPP 模拟出了车辆驾驶员行为策略随时间的变化，从而摆脱了只依赖历史信息进行长时间预测的局限性。最终，不同时刻的车辆编码会通过解码器转变为目标车辆的预测轨迹。

注意到后处理模块的输入是上一步中骨架网络生成的K模态场景预测。在测试或评估的时候，后处理模块会对所有的模态进行计算；而在训练的过程中，只会从中选择与真实场景最接近的 1 个模态计算损失函数和反向传播，这也是轨迹预测方法中对于多模态任务最常用的处理方式。受到某些图像降噪网络[42]的启发，轨迹后处理模块的输出会与原始的输入轨迹相加，共同组合为最终的预测轨迹，如图 2 所示。加入这种残差连接的结构是为了使后处理模块能够更关注于学习目标车辆受到的影响，从而得到更好的轨迹修正效果。4.2 的实验部分也展示了这个结构对模型性能的有效提升。接下来，在本节中将会依次介绍模型中车辆-车辆编码、车辆-车道编码和时序处理模块的结构。

由于本文的后处理模块主要基于注意力机制构建，因此在此会先对本文中注意力机制的计算方式和意义进行介绍。注意力机制的作用是根据不同元素的编码向量计算它们之间的交互关系。例如总共有N个编码向量，每个编码向量的维度为D，便可以通过注意力机制计算每个编码向量受到的其他$N-1$个向量的影响。从具体计算的角度来说，注意力机制共有 3 个输入，分别为查询矩阵，键矩阵和值矩阵，一般会将它们用字母$\boldsymbol{q}$、$\boldsymbol{k}$、$\boldsymbol{v}$代替。其函数可以写为如下的形式：

$$\boldsymbol{z} = \text{Attention}(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}). \quad (7)$$

仍然以上述的计算 N 个编码向量之间互相影响关系的问题为例，此时输入的$\boldsymbol{q}$矩阵，$\boldsymbol{k}$矩阵和$\boldsymbol{v}$矩阵都是维度为$[N, D]$的矩阵，即为N个维度为D的向量的堆叠。在注意力机制的函数内部，首先会将 3 个输入向量分别与不同的编码矩阵相乘：

$$\boldsymbol{Q}=\boldsymbol{q}\boldsymbol{W}^{\boldsymbol{Q}},\quad \boldsymbol{K}=\boldsymbol{k}\boldsymbol{W}^{\boldsymbol{K}},\quad \boldsymbol{V}=\boldsymbol{v}\boldsymbol{W}^{\boldsymbol{V}}, \tag{8}$$

其中，$\boldsymbol{W}^{\boldsymbol{Q}}$、$\boldsymbol{W}^{\boldsymbol{K}}$、$\boldsymbol{W}^{\boldsymbol{V}}$是维度为$[D,F]$，参数各不相同的矩阵，其作用为将输入向量从$D$维度升维到$F$维度。接着，通过$\boldsymbol{Q}$矩阵和$\boldsymbol{K}$矩阵的乘法计算不同元素施加影响的权重，将权重归一化后与$\boldsymbol{V}$矩阵相乘，得到其他元素对目标元素的影响编码：

$$\boldsymbol{m}=\sum_j \mathrm{softmax}\left(\frac{\boldsymbol{Q}}{\sqrt{F}}\boldsymbol{K}^{\boldsymbol{T}}\right)\boldsymbol{V}, \tag{9}$$

$\boldsymbol{m}$是维度为$[N,F]$的矩阵，该矩阵可以看作是N个维度为F的编码向量，而每个编码向量则代表了其他$N-1$个元素对当前元素的影响。为了获取当前元素受到影响后的状态，在注意力机制的最后会通过门函数的结构将影响编码$\boldsymbol{m}$与原始的输入向量进行综合：

$$\boldsymbol{g}=\mathrm{sigmoid}(\boldsymbol{W}^{\mathbf{gate}}[\boldsymbol{q},\boldsymbol{m}]), \tag{10}$$

$$\boldsymbol{z}=\boldsymbol{g}\odot\boldsymbol{q}+(1-\boldsymbol{g})\odot\boldsymbol{m}. \tag{11}$$

其中，$\boldsymbol{W}^{\mathbf{gate}}$为参数待训练的矩阵，$\odot$代表了 2 个矩阵之间每个元素的相乘。综上所示，注意力机制的作用主要在于构建不同元素编码之间的关联关系，计算受到其他元素影响后目标元素的编码。

车辆-车辆编码

这部分网络的主要目的是构建不同车辆在未来某个时间步上的交互，从而使得不同车辆的行为决策具有场景一致性。车辆-车辆编码器将会根据先前计算的车辆间交互关系对相应的车辆信息进行处理。在时间步t上，首先会以目标车辆在此时刻的速度向量计算其隐空间编码$\boldsymbol{z}_{\boldsymbol{i}}^{\boldsymbol{t}}$。假设在先前的计算中，认为车辆$j$对车辆$i$存在潜在的影响，那么会接着计算这个影响关系的编码$\boldsymbol{z}_{\boldsymbol{i},\boldsymbol{j}}^{\boldsymbol{t}}$：

$$\boldsymbol{z}_{\boldsymbol{i}}^{\boldsymbol{t}}=\phi_{\mathrm{center}}\left(\boldsymbol{R}_{\boldsymbol{i}}\left(\boldsymbol{x}_{\boldsymbol{i}}^{\boldsymbol{t}+\mathbf{1}}-\boldsymbol{x}_{\boldsymbol{i}}^{\boldsymbol{t}}\right)\right), \tag{12}$$

$$\boldsymbol{z}_{\boldsymbol{i},\boldsymbol{j}}^{\boldsymbol{t}_1}=\phi_{\mathrm{nbr}}\left(\left[\boldsymbol{R}_{\boldsymbol{i}}\left(\boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{t}_1+\mathbf{1}}-\boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{t}_1}\right),\boldsymbol{R}_{\boldsymbol{i}}\left(\boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{t}_1}-\boldsymbol{x}_{\boldsymbol{i}}^{\boldsymbol{t}}\right),t_1\right]\right),$$

$$t\le t_1<t+T, \tag{13}$$

其中ϕ_{center}和ϕ_{nbr}分别是 2 个参数不同的 MLP 网络，$\boldsymbol{R}_{\boldsymbol{i}}$是基于车辆$i$的面向构建的旋转矩阵，其目的是将所有的向量转换到车辆i的局部坐标系下。在建模车辆j对车辆i的潜在影响时，会对车辆j从时刻t到最后一个时间步$t+T$之间所有的轨迹向量进行计算，得到多个时刻的影响编码。相比于只使用时间t处的向量，这样做无疑能够更好地对车辆j的意图进行表示。同时，为了区分不同时刻上的编码，在计算时也加入了相对应的时间步信息t_1。

接着，会对$\boldsymbol{z}_{\boldsymbol{i}}^{\boldsymbol{t}}$和$\boldsymbol{z}_{\boldsymbol{i},\boldsymbol{j}}^{\boldsymbol{t}:\boldsymbol{T}}$使用注意力机制进行进一步的运算。由于这些编码存在不同时间和不同车辆这两个维度，最终编码器使用了双层的注意力机制来构建交互关系，如图 4 所示。首先利用注意力机制计算在t时刻车辆j对目标车辆i的影响向量$\hat{\boldsymbol{z}}_{\boldsymbol{i},\boldsymbol{j}}^{\boldsymbol{t}}$：

$$\hat{\boldsymbol{z}}_{\boldsymbol{i},\boldsymbol{j}}^{\boldsymbol{t}}=\mathrm{Attention}\left(\boldsymbol{z}_{\boldsymbol{i}}^{\boldsymbol{t}},\boldsymbol{z}_{\boldsymbol{i},\boldsymbol{j}}^{\boldsymbol{t}:\boldsymbol{T}},\boldsymbol{z}_{\boldsymbol{i},\boldsymbol{j}}^{\boldsymbol{t}:\boldsymbol{T}}\right), \tag{14}$$

对所有的周边车辆j进行计算后，再使用一层注意力机制的操作综合不同车辆的影响：

$$\hat{\boldsymbol{z}}_{\boldsymbol{i}}^{\boldsymbol{t}}=\mathrm{Attention}\left(\boldsymbol{z}_{\boldsymbol{i}}^{\boldsymbol{t}},\hat{\boldsymbol{z}}_{\boldsymbol{i},\boldsymbol{j}}^{\boldsymbol{t}},\hat{\boldsymbol{z}}_{\boldsymbol{i},\boldsymbol{j}}^{\boldsymbol{t}}\right), \tag{15}$$

经过车辆-车辆编码器的计算，场景中的N车都能各自得到T个融合了交互信息的编码向量，这些向量可以合写为$\hat{\boldsymbol{z}}_{\boldsymbol{1}:\boldsymbol{N}}^{\boldsymbol{t}:\boldsymbol{t}+\boldsymbol{T}}$。

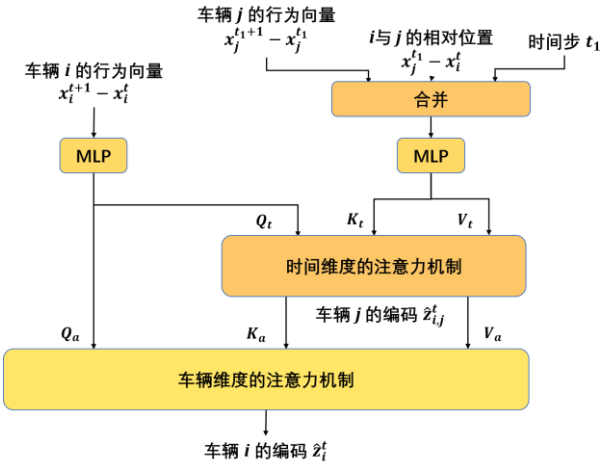


图 4 车辆-车辆编码中使用的双层注意力机制

Fig.4 The dual-layer attention mechanism used in vehicle-vehicle encoder

车辆-车道编码

由于驾驶员需要尽可能地保证车辆行驶在车道的中央，因此附近的车道信息同样会影响驾驶员的行为决策，车辆-车道编码器便会建模这种影响。与车辆之间的交互类似，车辆-车道编码器计算的是在时间步t时周围车道对目标车辆的影响。首先，将车道的位置信息进行坐标转换，并计算出代表潜在影响的编码：

$$\boldsymbol{z}_{i\xi}^{t}=\phi_{\text{lane}}\left(\left[\boldsymbol{R}_i\left(\boldsymbol{l}_{\xi}^{1}-\boldsymbol{l}_{\xi}^{0}\right),\boldsymbol{R}_i\left(\boldsymbol{l}_{\xi}^{0}-\boldsymbol{p}_i^{t}\right)\right]\right), \tag{16}$$

$\boldsymbol{R}_i$是以车辆i的角度为参数构造的旋转矩阵，$\boldsymbol{l}_{\xi}^{1}$和$\boldsymbol{l}_{\xi}^{0}$分别是车道ξ上第j个片段的结束位置和起始位置，ϕ_{lane}是 MLP 网络。后续的注意力机制的操作也与先前类似，只是计算 query 向量时使用的是车辆间交互模块输出的$\hat{z}_i^t$向量。最终，通过公式（13）、公式（14）、公式（15）计算注意力机制，获取车辆i在时间步t上的编码$\boldsymbol{h}_i^t\in\mathbb{R}^E$。

时序处理

经过 2 个编码器的计算，每辆车都得到了包含T个编码向量的序列$\boldsymbol{Seq}_i=\{\boldsymbol{h}_i^t\}_{t=t_0+1}^{t_0+T}$。这些编码向量经过车辆-车辆编码器和车辆-车道编码器的计算，与附近的车辆以及地图道路进行了充分的交互，代表不同时刻目标车辆根据当前环境信息做出的决策。由于不同的时刻的编码独立计算，它们彼此之间的信息可能存在着一定的差异或冲突，直接根据这样的编码序列生成轨迹可能导致最终性能的下降。为了建立编码序列中不同元素之间的联系，我们在后处理模块的最后添加了时序处理模块。该模块使用了注意力机制进行序列内的自相关运算，但在公式（9）的计算中，额外添加了 1 个掩模矩阵：

$$\boldsymbol{m}=\sum_j \text{softmax}\left(\frac{\boldsymbol{Q}}{\sqrt{F}}\boldsymbol{K}+\boldsymbol{M}_{ij}\right)\boldsymbol{V}, \tag{17}$$

$$\boldsymbol{M}_{ij}=\begin{cases}-\infty & \text{if } i<j;\\ 0 & \text{otherwise,}\end{cases} \tag{18}$$

添加掩模矩阵的目的是阻止序列中的元素与未来的信息发生交互，从而保证模块的计算具有因果性。最终，时序模块会输出目标车辆i在每个未来时间步上的信息编码序列$\widehat{\boldsymbol{Seq}}_i=\{\hat{\boldsymbol{h}}_i^t|\ t_0\le t\le T\}$。随后，后处理模块通过 MLP 网络依次将每个时刻的编码解码为相对应的车辆坐标，得到修正后目标车辆的未来预测轨迹。

注意到在我们的方法在解码的步骤中，每个时刻都会有 1 个对应的特征向量，而在自回归形式的解码器中[3, 26, 35-36]，目标车辆不同时刻上的预测位置也是基于不同的特征编码生成的。两者最大的不同在于，在自回归的形式中，每个特征编码只包含了过去时刻的信息；而我们的方法中，由于构建了附近附近未来轨迹的影响，因此目标车辆的所有特征编码都包含了周边车辆的未来意图，从而使得模型在生成未来轨迹的过程中能够对周边情况做出更好的判断。

在一些传统的轨迹预测方法中[6, 43]，类似的时序处理模块通常会先在输入序列$\boldsymbol{Seq}_i$的最后额外添加 1 个随机初始化的编码$\boldsymbol{h}_i^{t_0+T+1}$，完成计算后，取$\widehat{\boldsymbol{Seq}}_i$的最后一个编码作为下一个处理模块的输入。由于这些传统方法常常依据历史输入推断未来轨迹，因此模型的输入序列与输出序列之间没有时间步上的重叠。通过这种额外编码的方法，模型能够将历史不同时刻的信息都

整合到 1 个编码之中，进而方便后续模块对未来轨迹进行推断。与此不同的是，我们的后处理模块会根据未来的信息校正车辆的未来轨迹，这使得输入序列与输出序列处于相同的时间范围之内，并存在元素间的一一对应关系，因此我们选择将不同时间步上的信息分别用不同的编码进行表示。在实验部分也会对两种时序处理模块的结构进行对比，最终的结果将会在 4.2 节中进行展示。

2.3 模型训练

我们的框架整体可以分为 2 个部分，前端的边际预测模型，和后端的后处理模块。其中，边际预测模型使用的是预训练的骨架网络。我们对骨架网络的损失函数计算方式进行了一定的调整。一般而言，多模态的轨迹预测模型在训练时，每辆车只会选取与真实轨迹最贴近的 1 个模态计算损失函数并反向传播：

$$k = \operatorname{argmin}\left(\mathcal{L}_{\mathrm{FDE}}\left(\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{k}}, \bar{\boldsymbol{x}}_{\boldsymbol{n}}\right)\right), \tag{19}$$

其中，$\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{k}}$和$\bar{\boldsymbol{x}}_{\boldsymbol{n}}$分别是车辆$n$预测轨迹的第$k$个模态和车辆$n$的真实轨迹，评价与真实轨迹的贴近程度使用的是最终移位误差(FDE)的指标。而在我们的框架中，改为了对整个场景的模态进行选择：

$$k = \operatorname{argmin} \sum_{n=1}^{N} \mathcal{L}_{\mathrm{FDE}}\left(\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{k}}, \bar{\boldsymbol{x}}_{\boldsymbol{n}}\right). \tag{20}$$

第二部分的轨迹后处理模块以骨架模型的预测为输入，在训练模式下，只会选择与真实场景最贴切的 1 个模态结果用于后处理部分的训练。训练后处理模块的损失函数我们使用的是较为常见的 Huber Loss：

$$\mathcal{L} = \sum_{n=1}^{N} \sum_{t=1}^{T} \mathcal{L}_{\mathrm{L1}}\left(\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{t}}, \bar{\boldsymbol{x}}_{\boldsymbol{n}}^{\boldsymbol{t}}\right), \tag{21}$$

其中，$\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{t}}$和$\bar{\boldsymbol{x}}_{\boldsymbol{n}}^{\boldsymbol{t}}$分别是车辆 n 在时刻 t 的预测位置与真实位置。我们框架中的两部分模型会分开单独进行训练，在训练后处理模块时，前一部分的边际预测模型中的参数会被锁定。

3. 实验

3.1 实验设置

短时间多车预测任务 目前，Argoverse[44]数据集是轨迹预测领域中认可度较高，同时应用十分广泛的短时间预测数据集。因此，尽管 TPP 方法针对于长时间的预测任务设计，但我们仍然首先在 Argoverse 数据集上对

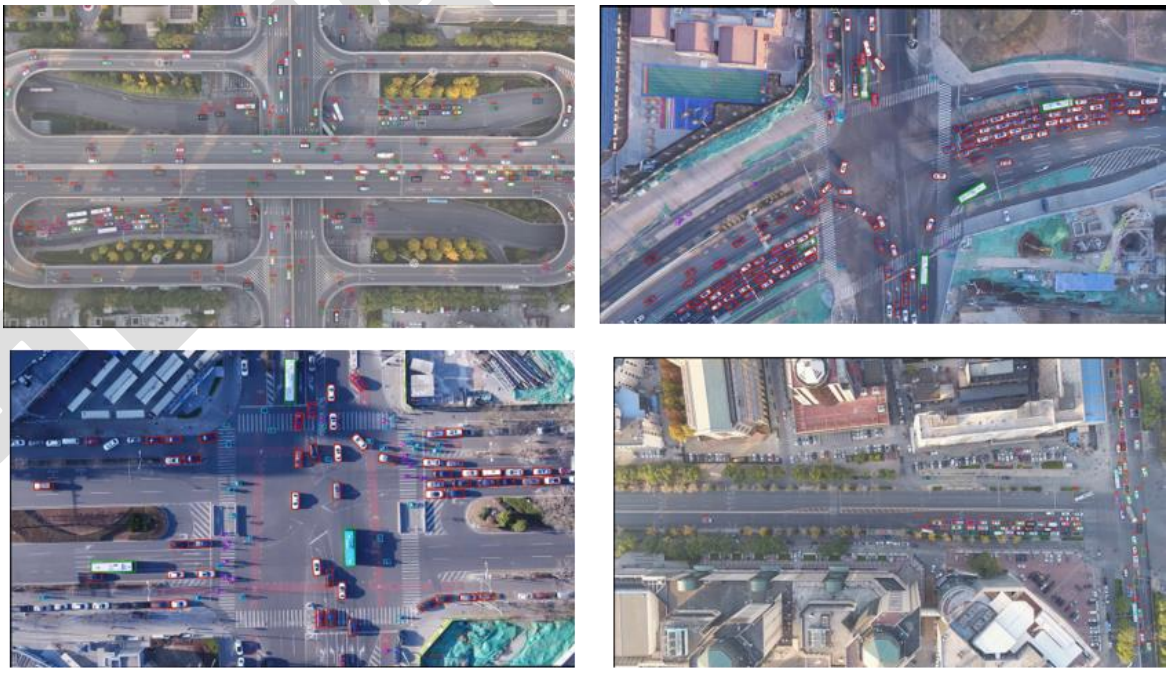


图 5 北京交通发展研究院的 TrafficHUT 数据库

Fig.5 The TrafficHUT Dataset from Beijing Transport Institute

TPP 方法进行评估，以此证明 TPP 中进行的改进不会降低模型在短时间预测任务中的性能，并能够维持有效的预测性能。Argoverse 中共计包含了 323557 个交通场景，其中训练集、验证集和测试集分别包含了 205942、39472 和 78143 个场景。数据集中所有的场景时长为 5s，采样频率为 10Hz。其中，前 2s 的数据会作为轨迹预测模型的输入，而模型需要预测目标车辆在剩余 3s 中的行动。

长时间多车预测任务 尽管 Argoverse 数据集应用广泛，但其中的预测任务并不能完全满足我们的需求。首先，每条轨迹的预测时长较短，无法体现出长时间预测任务中可能出现的各种问题；其次，Argoverse 的数据采集于迈阿密和匹兹堡这两座美国城市，而中国的道路情况和司机驾驶习惯与美国有很大的不同，为了使 TPP 后续能够更好地在中国的交通环境进行实际的应用，我们接着在 TrafficHUT 数据库[45]上对方法进行了评估。TrafficHUT 是由北京市交通发展研究院通过道路交通流航拍和人工智能轨迹识别技术制作的高精度标准化轨迹数据库，覆盖了 1200m 的道路长度，并具有 0.04s 时间精度和 0.5m 空间精度的车辆轨迹信息。其拍摄场景如图 5 所示。我们从 TrafficHUT 数据库中选取了 30min 的交叉口数据，包括车辆的轨迹信息以及高清的地图信息，并将连续的场景分割为了多个时长为 8s 的片段。考虑到交叉路口处存在诸多影响因素，例如遵守信号灯的规则，礼让行人等，为了简化建模的难度，数据集中去除了这些受到额外影响的轨迹。在 8s 的场景中，前 1s 的场景片段会作为输入，而轨迹预测模型需要输出后续 7s 的场景车辆轨迹。

评价指标 我们使用了常规的轨迹预测指标来评估我们的模型，包括了最小平均移位误差(minimum average displacement error, minADE)、最小最终移位误差(minimum final displacement error, minFDE)、丢失率(missing rate, MR)和碰撞率(collision rate, CR)。我们的模型会预测 6 个模态，度量指标 minADE 会计算表现最好的轨迹与真实轨迹之间的 l2 距离，而 minFDE 计算的是最优轨迹与真实轨迹的最后一个时刻的距离。MR 指的是预测轨迹终点与真实轨迹终点之间距离超过 2m 的概率。数据集的每个场景中都会指定 1 个特定的车辆，一般称为目标车辆。在计算碰撞概率时，每个场景中我们只会考虑目标车辆，统计场景中目标车辆与其他车辆发生了碰撞的次数占场景总数的比例。

3.2 定量分析

在短时间预测任务上的对比

我们首先在 Argoverse 数据集上对 TPP 进行了评估。由于 Argoverse 数据集没有提供车辆的形状信息，因此最终只对 minADE、minFDE 和 MR 这 3 个指标进行了对比，如表 1 所示。我们选择与一些其他文献中的经典算法[5,8]进行对比，它们分别是基于图表征和目标导向的经典预测方法。另外，我们还选择与 HiVT[6]方法进行对比，它是目前 Argoverse 数据集上的性能最好的算法之一。最终的结果说明我们的后处理方法要好于传统的基线算法，并且能够与 HiVT 方法性能持平。可以看到，尽管 TPP 模型的设计目标是解决长时间预测中车辆未来交互和策略更新的问题，但这些改进并不会牺牲模型在短时间预测任务上的性能。

表 1 在 Argoverse 数据集上结果的定量对比，最好的结果已加粗表示

Table 1 Quantitative comparison on Argoverse, the best result is highlighted in bold

	minADE↓	minFDE↓	MR↓
基于图的方法[7]	0.87	1.36	0.16
目标导向的方法[10]	0.76	1.04	**0.10**
HiVT[8]	**0.69**	1.05	**0.10**
Our Method	**0.69**	**1.02**	**0.10**

在长时间预测任务上的对比

接着，我们在 TrafficHUT 数据集上将 TPP 与目前性能最好的算法进行了对比。在 TrafficHUT 数据集中进行的是长时间的多车预测任务，我们选择了 HiVT[6] 作为对比的基线算法，结果如表 2 所示。可以看到，在 minADE、minFDE、MR 和 CR4 个指标上，我们的方法相较于基线算法都有较为显著的提升。其中，TPP 方法在 minADE 指标上获得了 48.7%的提升，在 CR 指标上获得了 37.0%的提升，这说明我们的模型能够提供更为准确的轨迹预测结果，并有效地缓解长时间预测中更容易出现的车辆碰撞的问题。而在 minFDE 和 MR 上，我们的模型分别取得了 51.3%和 30.1%的提升，这表示 TPP 方法对车辆最终位置的预测更加准确。综上所述，我们的 TPP 方法能够大幅度提升长时间预测任务中生成轨迹的质量。

表 2 在 TrafficHUT 数据集上结果的定量对比，最
Table 2 Quantitative comparison on In-house dataset, the best result is highlighted in bold

	minADE↓	minFDE↓	MR↓	CR↓
HiVT[8]	3.82	7.33	0.486	0.1452
TPP	**1.96**	**3.57**	**0.3395**	**0.0915**

消融实验

为了验证后处理网络中各个模块的作用，我们在 TrafficHUT 数据集上依次移除了模型中的一部分，并评估了模型删除模块后的性能。如表 3 所示，我们依次去除了模型中的车辆-车辆编码、车辆-车道编码、残差连接结构，并尝试了两种不同的时序处理结构。首先，去除车辆-车辆编码的部分，后处理网络将无法感知不同车辆之间在未来时刻的交互，从而无法避免车辆未来轨迹之间的碰撞；而去除车辆-车道编码的部分，后处理网络将无法纠正车辆未来时刻中偏离车道的行为；失去了这 2 个模块后，模型只能基于过去的过时信息推断未来轨迹，进而导致了性能的下降。除此之外，残差连接和时序处理结构也是模型中相当重要的部分。其中，残差连接指的是图 2 中后处理模块后方的原始预测轨迹数据的连接，该结构对模型的性能能够产生很大的影响。由此可以看出，间接预测环境施加给目标车辆未来轨迹的影响，取得的效果要好于传统方法中直接预测车辆的未来轨迹。两种不同的时序处理结构已经在 3.2 节中进行了介绍，其中，Temporal2 是将不同时刻信息整合到 1 个编码中的结构，Temporal1 是不同时刻分开编码的结构。从表中可以看出在后处理模块的输入输出模式下，采用 Temporal1 的结构计算序列中不同时间步之间的联系要比采用 Temporal2 的结构更加合理。

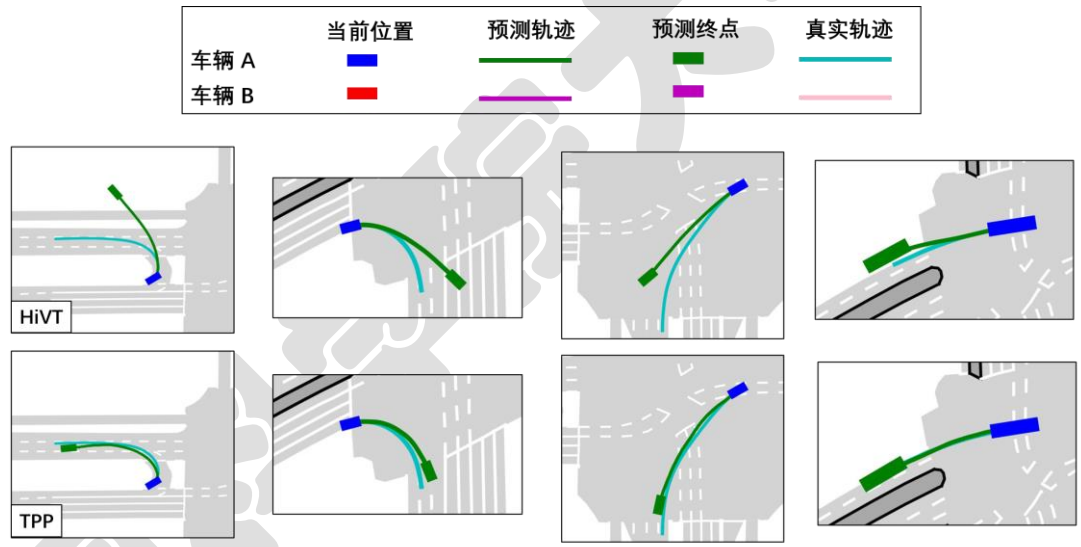
表 3 对网络中不同模块的消融实验，最好的结果已加粗表示

Table 3 Ablation study on different components, the best result is highlighted in bold

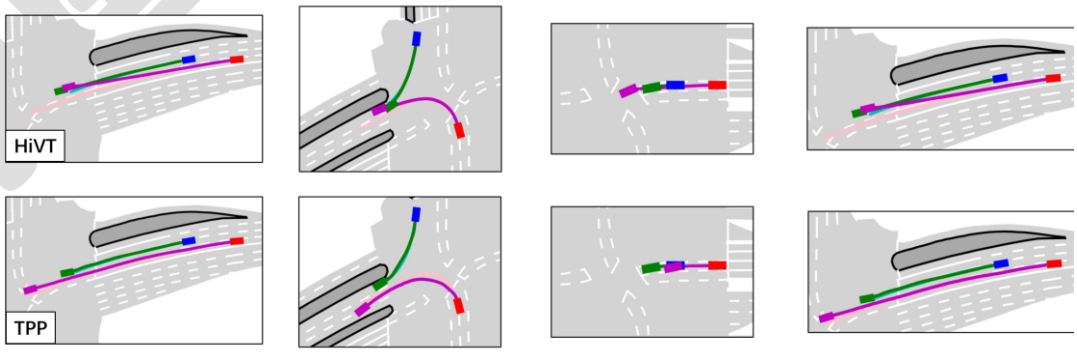
A-A	A-L	ShorCut	Temporal1	Temporal2	minADE	minFDE	MR	CR
	√	√	√		2.01	3.76	0.357	0.0987
√		√	√		1.98	3.61	0.343	0.0932
√	√		√		2.11	4.13	0.365	0.1176
√	√	√		√	2.08	4.23	0.362	0.1285
√	√	√	√		**1.96**	**3.57**	**0.3395**	**0.0915**

3.3 定性分析

在本节中，我们将通过可视化的方式，分别从偏离车道以及多车冲突这 2 个角度展示 TPP 方法的效果。在图 6 中，我们可视化了内部数据集上的多个场景，并将基线算法与我们的 TPP 方法进行了对比。其中，场景中车辆的初始位置会分别使用蓝色和红色方块表示，它们的预测轨迹以及最终位置则分别用绿色和紫色的线条与方块表示，它们在数据集中的真实轨迹则分别用浅蓝色和粉色的线条表示。在图 6(a)的几个场景下，传统基线算法的预测轨迹会出现驶出车道的情形。而我们的 TPP 方法因为能够利用到较远位置处的车道信息，因此能够及时地对车辆的行为策略做出调整，从而避免了偏离车道的问题。而在图 6(b)中，传统基线算法



（a） TPP 修正了车辆预测轨迹偏离车道的情形



（b） TPP 消除了多车预测轨迹之间的碰撞

图 6 TPP 方法与基线算法的可视化对比

Fig.6 Visual comparison between our TPP method and baseline model

预测的场景中出现了车辆换道不合时宜、转弯车辆间发生冲突、以及路口处车辆制动不及时的问题。需要说明的是，在实际的任务中模型需要对场景下所有的车辆轨迹进行预测，此处为了方便展示只可视化了发生冲突的两辆车。而在这些情形下，TPP 方法都能够通过后处理模块中的注意力机制，捕获到车辆之间在未来时刻上的交互，并对相应优先级更低的车辆轨迹做出准确的调整，最终保证预测出的多车轨迹具有场景一致性。

4. 结论

本文提出了一种全新的基于轨迹后处理的联合预测方法 TPP，它通过注意力机制的结构实现了多车场景下车辆间的交互，并利用后处理模块中单车多编码的特性缓解了以往联合预测方法在长时间预测中行为决策无法实时调整的问题。TPP 首先通过 1 个预训练的边际预测网络 HiVT 获取场景的初步预测，接着使用轨迹后处理模块进行修正，从而保证模型生成的预测具有场景一致性。在轨迹后处理模块中，首先通过车辆-车辆编码以及车辆-道路编码构建了不同时间步上车辆行为意图的交互，接着利用时序处理模块整合了机动车在不同时刻上的策略，实现了行为决策的实时调整。最终，我们通过实验证实了在 Argoverse 数据集和内部数据集上，TPP 方法已经取得了同类方法中最好的性能水平。

参考文献

[1] 徐杰,裴晓飞,杨波,等. 融合车辆轨迹预测的学习型自动驾驶决策[J].汽车安全与节能学报, 2022,13(2):317-324.DOI:10.3969/j.issn.1674-8484.2022.02.012.

[2] 王卫锋, 胡靖昊, 贺琰, 等. 出租车司机的多源轨迹同轨分析[J]. 中国科学院大学学报, 2023, 40(3): 313-321. DOI: 10.7523/j.ucas.2021.0078.

[3] Salzmann T, Ivanovic B, Chakravarty P, et al. Trajectron++: dynamically-feasible trajectory forecasting with heterogeneous data[C]//European Conference on Computer Vision. Glasgow: Springer, 2020: 683-700. DOI: 10.1007/978-3-030-58523-5_40.

[4] Gilles T, Sabatini S, Tsishkou D, et al. HOME: heatmap output for future motion estimation[C]//2021 IEEE International Intelligent Transportation Systems Conference (ITSC). Indianapolis, IN, USA. IEEE, 2021: 500-507. DOI: 10.1109/ITSC48978.2021.9564944.

[5] Liang M, Yang B, Hu R, et al. Learning lane graph representations for motion forecasting[C]// European Conference on Computer Vision. Glasgow: 2020: 541-556. DOI: 10.1007/978-3-030-58536-5_32.

[6] Zhou Z K, Ye L Y, Wang J P, et al. HiVT: hierarchical vector transformer for multi-agent motion prediction[C]//2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA. IEEE, 2022: 8813-8823. DOI: 10.1109/CVPR52688.2022.00862.

[7] Zhao H, Gao J Y, Lan T, et al. TNT: target-driveN trajectory prediction[C]//Conference on Robot Learning. PMLR, 2021: 895-904. https://proceedings.mlr.press/v155/zhao21b.html.

[8] Gu J R, Sun C, Zhao H. DenseTNT: End-to-end trajectory prediction from dense goal sets[C]// 2021 IEEE/CVF International Conference on Computer

Vision (ICCV). Montreal, QC, Canada. IEEE, 2021: 15283-15292. DOI: 10.1109/ICCV48922.2021.01502.

[9] Gilles T, Sabatini S, Tsishkou D, et al. GOHOME: Graph-oriented heatmap output for future motion estimation[C]//2022 International Conference on Robotics and Automation (ICRA).Philadelphia, PA, USA. IEEE, 2022: 9107-9114. DOI: 10.1109/ICRA46639.2022.9812253

[10] Gilles T, Sabatini S, Tsishkou D, et al. THOMAS: trajectory heatmap output with learned multi-agent sampling[EB/OL]. 2021: 2110.06607.（2021-10-13）[2022-10-13].http://arxiv.org/abs/2110.06607v3.

[11] Zhang Q C, Gao Y F, Zhang Y K, et al. TrajGen: Generating realistic and diverse trajectories with reactive and feasible agent behaviors for autonomous driving[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(12): 24474-24487 . DOI: 10.1109/TITS.2022.3202185

[12] Zeng W Y, Liang M, Liao R J, et al. LaneRCNN: Distributed representations for graph-centric motion forecasting[C]//2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Prague, Czech Republic. IEEE, 2021: 532-539. DOI: 10.1109/IROS51168.2021.9636035..

[13] Varadarajan B, Hefny A, Srivastava A, et al. MultiPath++: Efficient information fusion and trajectory aggregation for behavior prediction[C]//2022 International Conference on Robotics and Automation (ICRA). ACM, 2022: 7814-7821. DOI: 10.1109/ICRA46639.2022.9812107.

[14] Sun Q, Huang X, Gu J R, et al. M2I: from factored marginal trajectory prediction to interactive prediction[C]//2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA. IEEE, 2022: 6533-6542. DOI: 10.1109/CVPR52688.2022.00643.

[15] Rowe L, Ethier M, Dykhne E H, et al. FJMP: factorized joint multi-agent motion prediction over learned directed acyclic interaction graphs[C]//2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Vancouver, BC, Canada. IEEE, 2023: 13745-13755. DOI: 10.1109/CVPR52729.2023.01321.

[16] Li D, Zhang Q C, Lu S, et al. Conditional goal-oriented trajectory prediction for interacting vehicles[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, PP(99): 1-13. DOI: 10.1109/TNNLS.2023.3321564.

[17] Jiang C, Cornman A, Park C, et al. MotionDiffuser: controllable multi-agent motion prediction using diffusion[C]//2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Vancouver, BC, Canada. IEEE, 2023: 9644-9653. DOI: 10.1109/CVPR52729.2023.00930.

[18] Phan-Minh T, Grigore E C, Boulton F A, et al. Covernet: Multimodal behavior prediction using trajectory sets[C]// 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA. IEEE, 2020: 14062-14071. DOI: 10.1109/CVPR42600.2020.01408.

[19] Chen Y X, Ivanovic B, Pavone M. ScePT: Scene-consistent, policy-based trajectory predictions for

planning[C]// 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA. IEEE, 2022: 17082-17091. DOI: 10.1109/CVPR52688.2022.01659.

[20] Bi H K, Mao T L, Wang Z Q, et al. A deep learning-based framework for intersectional traffic simulation and editing[J]. IEEE Transactions on Visualization and Computer Graphics, 2020, 26(7): 2335-2348. DOI: 10.1109/TVCG.2018.2889834.

[21] Chai Y N, Sapp B, Bansal M, et al. MultiPath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction[EB/OL]. 2019: 1910.05449.(2019-10-12)[2022-09-14].http://arxiv.org/abs/1910.05449v1.

[22] Gao J Y, Sun C, Zhao H, et al. VectorNet: Encoding HD maps and agent dynamics from vectorized representation[C]// 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA. IEEE, 2020: 11522-11530. DOI: 10.1109/CVPR42600.2020.01154.

[23] Luo W J, Park C, Cornman A, et al. JFP: Joint future prediction with interactive multi-agent modeling for autonomous driving[C]//Conference on Robot Learning. PMLR, 2023: 1457-1467. DOI: 10.48550/arXiv.2212.08710.

[24] Cheng H, Liu M M, Chen L, et al. GATraj: A graph- and attention-based multi-agent trajectory prediction model[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2023, 205: 163-175. DOI: 10.1016/j.isprsjprs.2023.10.001

[25] 连静, 丁荣琪, 李琳辉, 等. 基于图模型和注意力机制的车辆轨迹预测方法[J]. 兵工学报, 2023, 44(7): 2162-2170. DOI: 10.12382/bgxb.2022.0117.

[26] Yuan Y, Weng X S, Ou Y L, et al. AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting[C]// 2021 IEEE/CVF International Conference on Computer Vision (ICCV). Montreal, QC, Canada. IEEE, 2021: 9793-9803. DOI: 10.1109/ICCV48922.2021.00967.

[27] Ngiam J, Caine B, Vasudevan V, et al. Scene Transformer: a unified architecture for predicting multiple agent trajectories[EB/OL]. 2021: 2106.08417.(2021-06-15)[2022-10-19].http://arxiv.org/abs/2106.08417v3.

[28] Liu Y C, Zhang J H, Fang L J, et al. Multimodal motion prediction with stacked transformers[C]// 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, TN, USA. IEEE, 2021: 7573-7582. DOI: 10.1109/CVPR46437.2021.00749

[29] Huang Z Y, Liu H C, Lv C. GameFormer: game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving[C]//2023 IEEE/CVF International Conference on Computer Vision (ICCV). Paris, France. IEEE, 2023: 3880-3890. DOI: 10.1109/ICCV51070.2023.00361.

[30] Shi S S, Jiang L, Dai D X, et al. MTR++: multi-agent motion prediction with symmetric scene modeling and guided intention querying[J]. IEEE Trans Pattern Anal Mach Intell, 2024, 46(5): 3955-3971. DOI: 10.1109/tpami.2024.3352811.

[31]李文礼,韩迪,石晓辉,等.基于时-空注意力机制的车辆轨迹预测[J].中国公路学报, 2023, 36(1):226-239.DOI:10.19721/j.cnki.1001-7372.2023.01.018.

[32] Pang Y T, Guo Z H, Zhuang B N. ProspectNet: weighted conditional attention for future interaction modeling in behavior prediction[EB/OL]. 2022: 2208.13848.(2022-08-29)[2022-11-04]. DOI:10.48550/arXiv.2208.13848.

[33]秦胜君,李婷.多交互车辆轨迹预测研究[J].计算机工程与应用, 2021, 57(11): 232-238.DOI:10.3778/j.issn.1002-8331.2005-0154.

[34]连静,李硕贤,刘一荻,等.基于车道目标引导的车辆轨迹预测[J].汽车工程, 2023, 45(8):1353-1361.DOI: 10.19562/j.chinasae.qcgc.2023.08.006

[35] Tang Y C, Salakhutdinov R. Multiple futures prediction[EB/OL] .2019: 10.48550/arXiv.1911.00997.(2019-11-04)[2022-09-05]. https://www.semanticscholar.org/reader/1928a851f7454223803d13e7260fa5f26979ffab.

[36] Seff A, Cera B, Chen D, et al. MotionLM: multi-agent motion forecasting as language modeling[C]//2023 IEEE/CVF International Conference on Computer Vision (ICCV). Paris, France. IEEE, 2023: 8545-8556. DOI: 10.1109/ICCV51070.2023.00788.

[37]季学武, 费聪, 何祥坤, 等. 基于LSTM网络的驾驶意图识别及车辆轨迹预测[J]. 中国公路学报, 2019, 32(6): 34-42. DOI: 10.19721/j.cnki.1001-7372.2019.06.003.

[38]吴晓建,危一华,王爱春,等.基于融合Dropout与注意力机制的LSTM-GRU车辆轨迹预测[J].湖南大学学报(自然科学版), 2023, 50(4):65-75.DOI: 10.16339/j.cnki.hdxbzkb.2023155

[39]张晓宁.基于LSTM网络的车辆轨迹预测研究[J].汽车实用技术, 2020, 45(22): 32-33, 39.DOI:10.16638/j.cnki.1671-7988.2020.22.011.

[40] Niedoba M, Lavington J, Liu Y P, et al. A diffusion-model of joint interactive navigation [EB/OL]. 2023: 2309.12508.(2023-09-21)[2023-10-30].http://arxiv.org/abs/2309.12508v2.

[41] Chang W J, Tang C, Li C R, et al. Editing driver character: Socially-controllable behavior generation for interactive traffic simulation[J]. IEEE Robotics and Automation Letters, 2023, 8(9): 5432-5439. DOI: 10.1109/LRA.2023.3291897.

[42] Tai Y, Yang J, Liu X M, et al. Memnet: A persistent memory network for image restoration[C]// 2017 IEEE International Conference on Computer Vision (ICCV). Venice, Italy. IEEE, 2017: 4549-4557. DOI: 10.1109/ICCV.2017.486.

[43] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding[EB/OL]. 2018: 1810.04805.(2018-10-11)[2023-11-20].http://arxiv.org/abs/1810.04805v2.

[44] Chang M F, Lambert J, Sangkloy P, et al. Argoverse: 3D tracking and forecasting with rich maps[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA. IEEE, 2019: 8740-8749. DOI:

10.1109/CVPR.2019.00895.

[45] 北京市交通发展研究院. TrafficHUT 交通轨迹库. [EB/OL]. 2023. [2023-11-25].http://www.traffichut.net/.