

# MHAttention

## 简介

在许多序列建模任务中（如自然语言处理或轨迹生成），Transformer 架构及其注意力机制已被证明非常有效。在轨迹优化任务中，我们可以利用多头注意力来高效融合来自不同来源的信息，特别是可以自组织的带有时空结构的信息，比如：车辆自身状态、环境中的障碍物信息以及目标点信息，。

## 什么是注意力

注意力机制允许模型在生成输出的每一部分时，聚焦于输入的不同部分。其核心思想包括：

- Query (Q):** 表示当前需要上下文信息的元素。
- Key (K):** 表示从中获取信息的元素。
- Value (V):** 包含用于更新 Query 的实际信息。

注意力函数的数学表达式为：

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V$$

其中  $d_k$  是 Key 的维度。

与单一注意力机制不同，多头注意力将 Q、K、V 分解为多个子空间（或“头”），允许模型同时捕捉多种关系。具体步骤包括：

- 线性投影：** 将 Q、K、V 投影到多个低维空间。
- 并行注意力：** 分别计算每个头的注意力。
- 拼接：** 将所有头的输出拼接在一起。
- 最终线性投影：** 将拼接后的输出再投影到所需的维度。

这种机制在处理复杂数据（比如混杂的障碍物信息）时尤为有用，因为它允许模型同时考虑多种交互关系。

轨迹拟合涉及为车辆规划最佳路线或运动序列，同时需要考虑：

- 动态状态：** 车辆当前的速度、航向、加速度等信息。

- **环境障碍：** 用各自特征表示的静态或动态障碍物。
- **目标条件：** 表示目标位置或目的地。

拓扑信息表征上的挑战：

- **时序依赖：** 捕捉历史信息以预测未来运动。
- **空间关系：** 理解车辆、障碍物与目标之间的相对位置关系。

在进行的case中，数据集包含：

- **自身信息：**
  - **Token：** 代表车辆位置的唯一标识符。
  - **速度测量：** 包括航向、速度、加速度等。
- **障碍物信息：**
  - **Token：** 代表Agent位置的唯一标识符。
  - **相对位置：** 使用position embedding进行embedding
  - **速度以及附加特征：** 包括航向、速度、加速度、类型等。
- **目标点：**
  - **Token：** 代表自车目标位置的唯一标识符。
  - **速度测量：** 包括航向、速度、加速度等。

## Encoder-Decoder 架构

我们的模型采用：

- **Encoder：** 使用混合多头注意力机制处理数据的拓扑结构。
  - 从自身信息中生成 Q（经过对速度信息的加权注意力）。
  - 从障碍物信息+目标点信息+障碍物生成 K, V。目标点代表车辆期望达到的状态，其特征包含了未来的位置信息和动态指标，即便存在时空差（例如目标状态可能延后约2秒），目标点仍能为生成过程提供明确的目标。
- **Decoder：** 使用“Decoder only”方法，其中：
  - Decoder 的 Query 来自于当前生成过程中的历史信息。具体来说，Decoder 会对已生成的轨迹或中间状态进行嵌入，形成查询向量，该向量反映了当前生成的上下文。通过与 Encoder 固定的 KV（融合后的全局上下文信息）进行注意力计算，Decoder 能够根据历史信息和全局上下文确定下一步的生成输出。

- Encoder 输出的 KV 作为固定的上下文信息被传递给 Decoder，使得 Decoder 在生成过程中可以参考全局的时空信息，从而生成连贯且符合目标要求的轨迹。
- Encoder 输出的 KV 作为固定的上下文信息。