SIMPL: 自律走行のためのシンプルで効率的なマルチエージェント動作予測ベースライン

Lu Zhang[1], Peiliang Li[2], Sikang Liu[2], and Shaojie Shen[1]

概要- 本論文では、自律走行車のためのシンプルで効率的な動き予測ベースライン(SIMPL)を紹介する。高精度でありながら反復計算を行う従来のエージェント中心手法や、精度と汎用性が損なわれたシーン中心手法とは異なり、SIMPLは関連するすべての交通参加者に対して、リアルタイムで正確な動き予測を提供する。精度と推論速度の両方の向上を達成するために、我々は、対称的な方法で有向メッセージパッシングを実行するコンパクトで効率的なグローバル特徴融合モジュールを提案し、ネットワークが単一のフィードフォワードパスですべての道路ユーザーの将来の動きを予測することを可能にし、視点シフトによる精度損失を緩和する。さらに、軌道解読におけるベルンシュタイン基底多項式を用いた連続軌道パラメタリゼーションを検討し、下流の計画タスクに有用な、任意の時点における状態とその高次導関数の評価を可能にする。強力なベースラインとして、SIMPLはArgoverse 1 & 2モーション予測ベンチマークにおいて、他の最先端手法と比較して高い競争力を示す。さらに、SIMPLは軽量で推論レイテンシが小さいため、拡張性が高く、実世界での搭載が期待できる。https: //github.com/HKUST-空中ロボティクス/SIMPL でオープンソース化。

I. Introduction

自律走行車、特に下流の意思決定・計画モジュールでは、正確でタイムリーな意図と軌道予測が安全性と走行快適性の両方に大きく寄与するため、周囲の交通参加者の運動予測は不可欠である。

学習ベースの動き予測にとって、最も重要なトピックの1つはコンテキスト表現である。初期のアプローチでは、周囲のシーンをマルチチャンネルの鳥瞰画像として表現するのが一般的であった[1]-[4]。これに対して、最近の研究では、ベクトル化されたシーン表現[5]-[13]がますます採用されるようになってきており、位置や形状は点集合や地理座標を持つポリラインを用いて注釈付けされ、忠実度の向上や受容野の拡大につながる。しかし、ラスタライズド表現とベクトル化表現の両方において、重要な疑問が存在する。一つの簡単な方法は、自律走行車を中心とした座標系など、共有座標系(シーン中心)内のすべてのインスタンスを描き、その座標を入力特徴として直接利用することである。これにより、1回のフィードフォワードパスで複数のターゲットエージェントの予測を行うことができる[8, 14]。

L. ZhangとS. Shenは香港科学技術大学電子・コンピュータ工学科に所属している(email: {lzhangbz, eeshaojie}@ust.hk)。[2] P. LiとS. LiuはDJI Technology Company, Ltd., Shenzhen, Chinaに所属している(email: {peiliang.li, sikang.liu}@dji.com)。共著者 Lu Zhang.この研究は、香港博士号取得者の一部支援を受けた。フェローシップ・スキーム、一部HKUST-DJI共同イノベーション研究所によるもの。

図1:複雑な運転シナリオにおけるマルチエージェントの動作予測の説明図。我々の手法は、すべての関連するエージェントに対して同時に合理的な仮説をリアルタイムで生成することができる。自車両とその他の車両は、それぞれ赤と青で示されている。予測された軌跡は、タイムスタンプに応じてグラデーションカラーで可視化される。その他の例については添付のビデオを参照してください。

しかし、入力としてグローバル座標を使用すると、多くの場合大きなスパンで変化するため、タスク固有の複雑さが大幅に強化され、その結果、ネットワーク性能が低下し、新しいシナリオへの適応性が制限される。より良い精度とロバスト性を達成するために、一般的な解決策は、ターゲットエージェントの現在の状態に対してシーンコンテキストを正規化することである[5, 7, 10]-[13](エージェント中心)。これは、正規化処理と特徴エンコーディングを各ターゲットエージェントに対して繰り返し実行する必要があることを意味し、より良いパフォーマンスにつながるが、計算量は冗長になる。したがって、視点の変化に対する頑健性を保持したまま、複数のターゲットの特徴を効率的に符号化できるアプローチを探求することが不可欠である。

意思決定や運動計画など、下流の運動予測モジュールについては、将来の位置だけでなく、方位、速度、その他の高次導関数も考慮することが必須である。例えば、予測される周辺車両の方位は、安全でロバストな運動計画を確保するための重要な要素である、将来の時空間占有率を形成する上で極めて重要な役割を果たす[15, 16]。さらに、物理的制約に従わずに高次の量を独立に予測することは、予測結果に矛盾をもたらす可能性がある[17, 18]。例えば、速度がゼロであるにもかかわらず、位置の変位が発生する可能性があり、計画モジュールの混乱につながる。

本論文では、実世界のオンボードアプリケーションのためのマルチエージェント軌道予測における重要な問題に対処する、自律走行システムのためのSIMPL(Simple and efficient Motion Prediction baseLine)を提案する。まず、インスタンス中心のシーン表現に続いて、対称融合変換器(SFT)を導入することで、視点不変の特性によってもたらされる精度とロバスト性を保持しながら、1回のフィードフォワードパスで全てのエージェントの効率的な軌跡予測を可能にする。

対称コンテキストフュージョンに基づく他の最近の研究[19]-[21]と比較して、提案されたSFTは、よりシンプルで、より軽量で、実装が容易であり、オンボード展開に適している。

次に、ベルンシュタイン基底多項式(ベジエ曲線とも呼ばれる)に基づく、予測軌道の新しいパラメタリゼーション法を紹介する。この連続表現は平滑性を保証し、任意の時点における厳密な状態とその高次導関数の楽な評価を可能にする。我々の実証研究は、ベジエ曲線の制御点を予測する学習は、単項基底多項式の係数を推定する学習と比較して、より効果的で数値的に安定であることを示している。

最後に、提案されたコンポーネントは、シンプルで効率的なモデルにうまく統合されている。提案手法を2つの大規模モーション予測データセット[22, 23]で評価した結果、SIMPLは合理的な設計にもかかわらず、他の最先端手法と比較して高い競争力を持つことが実験結果から示された。さらに重要なことは、SIMPLは、定量的な性能を犠牲にすることなく、より少ない学習可能なパラメータとより低い推論レイテンシで、効率的なマルチエージェント軌道予測を達成し、実世界のオンボード展開に有望であるということである。また、SIMPLは強力なベースラインとして優れた拡張性を獲得することを強調する。簡潔なアーキテクチャは、モーション予測における最近の進歩とのストレートな統合を容易にし、全体的なパフォーマンスをさらに向上させる機会を提供する。

II. Related Work

A. コンテキストの符号化と融合

運転コンテキストは、大きく分けて、周囲のエージェントの過去の軌跡と静的マップ情報の2種類に分類される。時系列データとしての軌跡は、通常、時間ネットワークによって符号化される[24, 25]。地図特徴に関しては、初期の研究では、異なる意味要素を異なるチャンネルでレンダリングしたマルチチャンネルの鳥瞰画像として表現するのが一般的であり、その後、畳み込みニューラルネットワーク(CNN)を利用して特徴融合を行う[1]-[4]。しかし、ラスタライズは必然的に情報損失を導入し、限られた受容野をもたらす。これらの問題を解決するために、ベクトル化ベースの手法が提案され[5, 6]、ますます普及している[7]-[13]。このような手法では、マップ要素はポリライン[5, 7, 9, 11]やスパースグラフ[6, 10, 13]として表現され、生の座標を用いて空間情報を保持する。これらの特徴は、グラフニューラルネットワーク[26, 27]やTransformers[28]を介してさらに処理され、より高い忠実度と効率をもたらす。

B. 対称シーンモデリング

シーン中心[8, 14]とエージェント中心[5, 7, 10]-[13]の両表現には限界があり、精度と計算オーバヘッドのトレードオフが必要である。最近、特徴融合処理に対称モデリングを導入することで、この問題に対処するいくつかのアプローチ[9, 19]-[21]が登場した。HiVT[9]は、各エージェントのローカルコンテキストを正規化し、ローカルとグローバルの両方の特徴フュージョンに相対的なポーズを明示的に組み込むことで、この手法を視点不変にする。HDGT[19]とGoRela[20]は、異種グラフのメッセージパッシングにペアワイズ相対位置エンコーディングを導入している。さらに一歩進んで、QCNet[21]は、時間次元を相対位置エンコーディングに組み込むことで、視点不変の特性を空間-時間領域に拡張し、ストリーミング処理のサポートを可能にしている。これらのアプローチと比較して、我々の研究も同様のアイデアを採用しているが、よりシンプルで軽量、かつ実装が容易なコンパクトな対称特徴融合モジュールを提案している。

C. 軌道表現

予測される軌道は、位置座標[5, 6]や確率分布の混合[3, 9]のような離散的な状態のシーケンスとして表現されるのが一般的である。離散的な状態の間には明示的な制約がないため、これは常にギザギザの運動学的に実行不可能な軌道を導く。別の手法として、制御信号を予測し、運動学モデルに従って軌道に再帰的に統合する方法がある[29, 30]。しかし、このリカレント定式化は効率が悪く、知覚エラーの影響を受けやすい傾向がある。移動ロボットの軌道計画には、ベジエ曲線などの連続軌道パラメタリゼーションが広く用いられている。ある目的と制約を考慮しながら制御点を操作することで、滑らかで連続的な最適軌道を効率的に生成することができる[31, 32]。本論文では、出力形式としてベジエ曲線を活用し、単項基底多項式[17]と比較して優れた数値安定性を維持しながら、リカレントアンローリングなしでシングルステップデコーディングを保証する。

III. Methodology

A. 問題の定式化

軌道予測タスクは、観測された移動物体の運動履歴と周囲の地図情報に基づいて、ターゲットエージェントの将来の潜在的な軌道を生成することを含む。具体的には、N_a個の移動エージェント(AVを含む)がいる運転シナリオでは、マップ情報をMで表現し、$X = \{x_0, .. x_{Na}\}$ を用いて、全エージェントの観測された軌跡を総称する。ここで、各$x_i = \{x_{i,-H+1}, .. x_{i,0}\}$は、過去H回の時間ステップにおけるi番目のエージェントの過去の軌跡を表す。一般性を損なうことなく、マルチエージェント運動予測器は、シーン内の全てのN_aエージェントの潜在的な将来の軌道を生成し、$Y = \{y_0, .. y_{Na}\}$ と表される。各エージェントiについて、K個の可能な将来の軌跡とそれに対応する確率スコアが予測され、固有のマルチモーダル分布を捕捉する。マルチモーダル軌道は $y_i = \{y_i^1, .. y_i^K\}$ と表し、各 $y_i^k = \{y_{i,1}^k, .. y_{i,T}^k\}$ は予測地平Tにおけるi番目のエージェントのk番目の予測軌跡であり、確率スコアリストは$\alpha_i = \{\alpha_i^1, .. \alpha_i^K\}$ と表される。したがって、エージェントiのマルチモーダル軌道予測は、混合分布を推定しているとみなすことができる

$$P(\mathbf{y}_i|\mathbf{X}, \mathcal{M}) = \sum_{k=1}^{K} \alpha_i^k P(\mathbf{y}_i^k|\mathbf{X}, \mathcal{M}).$$

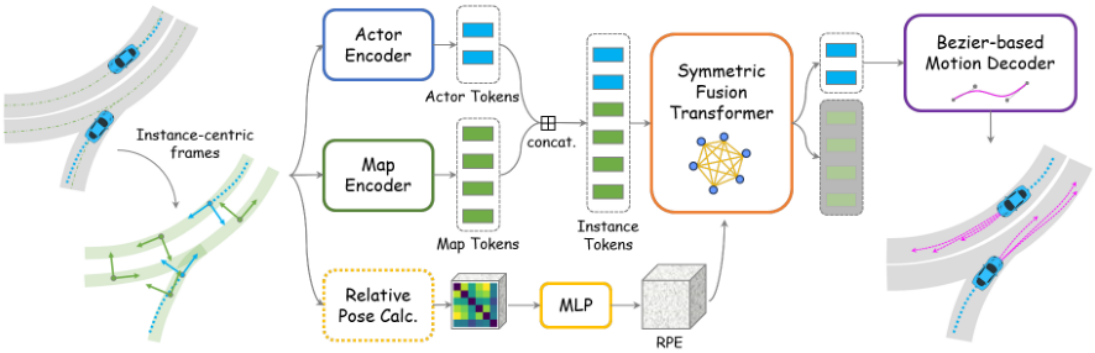

図2:SIMPLの説明図。最も単純なネットワークアーキテクチャを利用し、その有効性を実証している。意味インスタンスの局所特徴は単純なエンコーダで処理され、インスタンス間の特徴は相対位置埋め込みで保存される。提案する対称特徴変換器の後、モーションデコーダによりマルチモーダル軌跡予測結果を生成する。

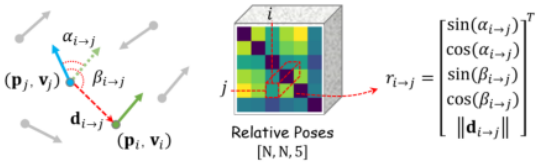

図3: 相対ポーズ計算の説明図。典型的なシーンが左側に描かれており、簡潔にするためにアンカーポーズのY軸を省いている。インスタンスiとjの間の相対姿勢は、方位差$\alpha_{i\to j}$、相対方位角$\beta_{i\to j}$、位置距離$\Delta d_{i\to j}$ Δで記述できる。全相対ポーズを計算し、3Dアレイとして定式化する。

本論文では主に限界運動予測に焦点を当てるが、シーンレベルの損失関数[8, 33]を含むことで、我々のアプローチを関節運動予測タスクにスムーズに拡張できることに注意。今後の重要な課題として残す。

B. フレームワークの概要

提案するSIMPLフレームワークの概要を図2に示す。まず、ベクトル化されたシーン表現を採用する。軌跡や車線セグメントなどの各意味インスタンスに対して、固有の特徴とインスタンス間の相対情報を分離するためのローカル参照フレームを構築する。次に、アクターとマップの特徴を単純なエンコーダで抽出し、インスタンスの相対的なポーズをペアで計算し、さらに多層パーセプトロン(MLP)でエンコードして相対位置埋め込み(RPE)を得る。インスタンストークンとRPEは、次に、対称的に特徴を更新するコンパクトで簡潔な融合モジュールである、提案された対称融合変換器(SFT)に送られる。最後に、ベジエ曲線のパラメータ化された軌道は、すべてのターゲットエージェントに対して同時に単純なデコーダによって予測される。

C. インスタンス中心のシーン表現

シーン中心の表現とは別に、シナリオはインスタンスのローカルフレームの下で、それらの間の相対的なポーズとともにベクトル化された特徴によって表現することができる。各意味要素の空間属性を正規化するために、ローカル参照フレームが確立され、これを「インスタンス中心」と呼ぶ。一般性を損なうことなく、エージェントの過去の軌跡について、現在の観測状態における参照フレームを特定する。車線セグメントなどの静的地図要素については、ポリラインの重心をアンカーポイントとし、終点間の変位ベクトルをヘディング角として採用する。

直感的には、ローカル座標フレームはインスタンスの「アンカーポーズ」とみなすことができ、したがって、相対的な空間情報はペアごとに簡単に計算することができる。

具体的には、要素iのグローバル座標フレーム下でのアンカーポーズは、その位置$p_i \in R^2$と方位ベクトル$v_i \in R^2$を用いて表現できる。20]に従い、要素iと要素jの相対姿勢を、方位差$\alpha_{i\to j}$、相対方位$\beta_{i\to j}$、距離$\Delta d_{i\to j}$ Δの3つの量を用いて記述する。数値的な安定性を高めるため、角度はサイン値とコサイン値で表現している。方位差$\alpha_{i\to j}$を次のように表す。

$$\sin(\alpha_{i\to j}) = \frac{\mathbf{v}_i \times \mathbf{v}_j}{\|\mathbf{v}_i\|\|\mathbf{v}_j\|}, \quad \cos(\alpha_{i\to j}) = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\|\|\mathbf{v}_j\|},$$

とし、相対方位角$\beta_{i\to j}$(変位ベクトル$d_{i\to j} = p_i - p_j$と方位ベクトルv_jのなす角)を次のように求める。

$$\sin(\beta_{i\to j}) = \frac{\mathbf{d}_{i\to j} \times \mathbf{v}_j}{\|\mathbf{d}_{i\to j}\|\|\mathbf{v}_j\|}, \quad \cos(\beta_{i\to j}) = \frac{\mathbf{d}_{i\to j} \cdot \mathbf{v}_j}{\|\mathbf{d}_{i\to j}\|\|\mathbf{v}_j\|}.$$

簡単のため、[20]で用いた距離値の位置符号化処理を省略し、相対空間情報を5次元ベクトル$r_{i\to j} = [\sin(\alpha_{i\to j}), \cos(\alpha_{i\to j}), \sin(\beta_{i\to j}), \cos(\beta_{i\to j}), \Delta d_{i\to j}\ \Delta]$とする。PyTorchやNumPyの放送機構を活用することで、全対全の相対空間情報を簡便に計算することができる。その結果、シーンが$N = N_a + N_m$個の意味要素を含むとすると、結果として得られる相対位置情報は[N, N, 5]の形状を持つ配列となり、$r_{i\to j}$はj番目の行とi番目の列に位置する。相対姿勢の計算の説明図を図 3 に示す。

D. コンテキスト特徴の符号化

インスタンス中心表現とインスタンスの相対位置エンコーディングを得た後、対応するエンコーダ(「トーケナイザ」としても機能する)を利用して、特徴ベクトルに変換する。SIMPLをシンプルにするために、過去の軌跡を扱うために1次元CNNベースのネットワーク[6]を使用し、静的な地図特徴を抽出するためにPointNetベースのエンコーダ[5, 34]を採用する。一般性を損なわない範囲で、全ての潜在特徴量にD個のチャンネルを持たせる。したがって、結果として得られるアクタートークンとマップトークンは、[N_a, D]と[N_m, D]の形状を持ち、N_aはアクターの数、N_mはマップ要素の数である。詳細な実装については、[5, 6]を参照されたい。さらに、相対的なポーズ

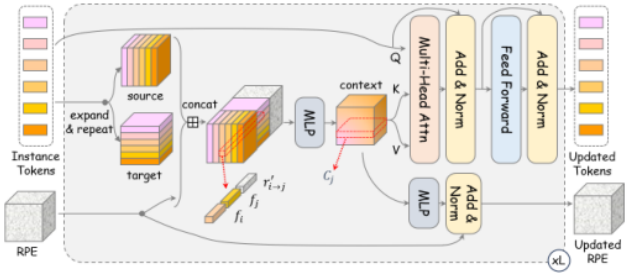

図4:L層を持つ提案する対称融合変換器(SFT)の説明図。インスタンストークンとRPEは各SFT層で再帰的に更新される。

符号化はさらにMLPによって符号化され、[N, N, D]の形状を持つ相対位置埋め込み(RPE)が得られる。

E. 対称融合変換器

インスタンストークンとそれに対応するRPEが得られたら、提案する対称融合変換器(SFT)を用いて、視点不変の方法でインスタンストークンを更新する。図4は、標準的なTransformer[28]と同様に、複数のSFT層を積み重ねた提案SFTの全体構造を示している。要するに、運転シーンを、入力インスタンス中心の特徴がノードとなり、RPEがエッジ情報を描写する、自己ループを持つ完全なダイグラフとみなすことができる。更新処理中、ノードの特徴はターゲットノードに関連するグラフエッジのみに影響され、特徴融合が視点不変のままであることを保証する。

微視的な見方では、i番目とj番目のインスタンスのトークンをそれぞれf_iとf_jとする。そして、f_iからf_jへの辺に関連するRPEベクトルを$r'_{i\to j}$とする。タプルf_i, f_j, $r'_{i\to j}$はノードiからノードjへの送信を意図した全ての情報を包含するので、これらの特徴を符号化するために単純なMLPを採用し、ノードjのi番目のコンテキストベクトルを得ることができる。

$$c_{i\to j} = \phi\left(f_i \boxplus f_j \boxplus r'_{i\to j}\right),$$

ここで、*は連結演算子を表し、$\phi : R^{3D} \to R^{D}$ は線形層、層正規化、ReLU活性化からなるMLPを表す。次に、ターゲットノードとそのコンテキストに対してクロスアテンションを行う、

$$f'_j = \mathrm{MHA}(\mathrm{Query}: f_j, \mathrm{Key}: C_j, \mathrm{Value}: C_j),$$

ここで、MHA(-, -, -)は標準的な多頭注意関数であり、$C_j = \{c_{i\to j}\}_{i\in\{1,\cdots,N\}}$ はトークンjの文脈ベクトルの集合である。C_jには$c_{j\to j}$も含まれ、各ノードの自己ループが存在することを示す。標準的なTransformerと同様に、注意メカニズムの後にポイントワイズフィードフォワード層が統合される。さらに、各層で$r'_{i\to j}$は、別のMLPを用いてコンテキストベクトルを再エンコードすることで更新され、その後に、残差接続によってそれぞれRPEに追加される。実際には、前述の特徴量融合をパラレル化することで、より効率的に実装することができます(図4参照)。まず、入力インスタンストークンF∈$R^{N\times D}$が与えられたとき、それを異なる次元に沿って展開し、N回複製してソースノードとターゲットノードの配列を構築し、両者とも[N, N, D]の形状を示す。連結後

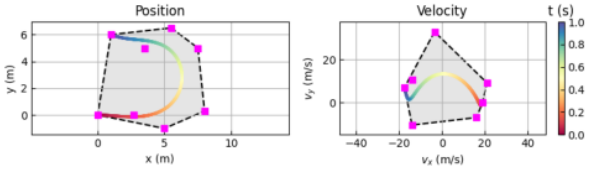

図5:2次元敗血症ベジエ曲線(左)。ピンクの点は制御点、グレーのポリゴンは対応する凸包である。軌跡の時間幅が1秒の場合、1次微分はまさに速度プロファイル(右)となり、これもホドグラフの性質上ベジエ曲線となる。

ソース配列、ターゲット配列、対応するRPEのタプルf_i, f_j, $r'_{i\to j}$の配列が得られ、ϕを適用してコンテキスト配列C∈$R^{N\times N\times D}$を得る。Cのj行目はちょうどC_jであり、トークンjを中心としたコンテキスト特徴の集合を表すことに注意。したがって、キーと値にはCを採用し、拡張F∈$R^{N\times 1\times D}$をクエリとする。次に、標準的なマルチヘッド注意モジュールは、コンテキスト特徴からインスタンストークンにメッセージを渡す。残りのSFT層もベクトル化された実装を享受しているが、その単純さゆえに詳細は掘り下げない。我々の提案するSFT層は、最近の「クエリ中心」手法[21, 35]と類似しているが、我々はグローバルな注意とRPE更新を取り入れ、よりコンパクトな設計を実現していることは注目に値する。詳細な実装については、公開されているコードを参照してください。

F. マルチモーダル連続軌跡デコーダ

対称的な大域的特徴融合の後、更新されたアクタートークンが集められ、マルチモーダルモーションデコーダに送られ、全エージェントの予測を生成する。ここでは、K個の可能な未来を予測し、各モードに対して、軌跡の回帰ヘッドと、対応する確率スコアのソフトマックス関数に続く分類ヘッドを持つ単純なMLPを適用する。

軌道回帰ヘッドに関しては、将来の軌道の位置を直接予測する従来のアプローチとは対照的に、我々は連続的なパラメータ化された表現を使用することにした。パラメータ化された曲線(例えば多項式)は連続的な表現をもたらし、任意の時点における滑らかな動きと厳密な高次導関数を得ることを可能にする。しかし、先行研究[30]によれば、単項基底多項式表現は性能を大きく低下させる。我々は、予測された係数の数値的な不均衡(詳細はセクションIV-C.2を参照)に退化のせいにし、回帰を難しいタスクにする。

性能低下を避けつつパラメトリック軌道の利点を活用するために、Bernstein基底多項式(すなわち、ベジエ曲線)を導入し、その係数を具体的な空間的意味を持つ制御点とすることで、より良い収束を実現する。具体的には、次数nのベジエ曲線は次のように書かれる。

$$f(t) = \sum_{i=0}^{n} b_n^i(t) p_i = \sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} p_i,\ t \in [0,1],$$

ここで、(t)のbはi次のベルンシュタイン基底、niは二項係数、tはパラメトリック曲線の変数、p_iは制御点である。n次のベジエ曲線の場合、全部でn + 1個の制御点があり、最初と最後の制御点は常に曲線の端点であることに注意(図5参照)。

ベジエ曲線はt∈[0, 1]上で定義されるので、実際の時間$\tau \in [0, \tau_{max}]$を正規化するので、パラメトリック曲線は$(\frac{\tau}{\tau_{max}})p_i$においてf（t）$=\sum_{i=0}^{n}$ bと書ける。さらに、ホドグラフの性質上、n次のベジエ曲線の微分は、制御点が$p_i^{(1)} = n(p_{i+1} - p_i)$で定義されるベジエ曲線のままであり、すなわち、軌道の速度プロファイルは次式で計算できる。

$$f'(t) = \frac{n}{\tau_{max}} \sum_{i=0}^{n-1} b_{n-1}^{i}(\frac{\tau}{\tau_{max}})(p_{i+1} - p_i),\ \tau \in [0, \tau_{max}].$$

実際には、融合されたアクター特徴から制御点へのマッピングを行う回帰ヘッドとして、単純なMLPを使用する。次に、各予測軌道$Y_{pos} \in R^{T\times 2}$の位置座標は、定数基底行列$B \in R^{T\times(n+1)}$と対応する予測2次元制御点$P \in R^{(n+1)\times 2}$(x軸とy軸は独立)を掛け合わせることで簡単に計算できる、

$$\begin{aligned}\mathbf{Y}_{pos} &= \mathbf{B} \times \mathbf{P} \\ &= \begin{bmatrix} b_n^0(t_1) & b_n^1(t_1) & \dots & b_n^n(t_1) \\ \vdots & \vdots & & \vdots \\ b_n^0(t_T) & b_n^1(t_T) & \dots & b_n^n(t_T) \end{bmatrix} \begin{bmatrix} p_0^x & p_0^y \\ \vdots & \vdots \\ p_n^x & p_n^y \end{bmatrix},\end{aligned}$$

ここで、Tは予測された軌跡に必要なサンプリングされたタイムスタンプの数であり、$t_i = \frac{\tau_i}{\tau_{max}}$は正規化された時点である。また、予測された軌道の速度やその他の高次導関数は、上記と同様の手順で求めることができることを指摘し、簡潔にするために省略する。車両や自転車のような非ホロノミック制約を持つエージェントの場合、ヨー角は軌跡の接線ベクトルと一致し、速度推定から各状態の方位角を導出することができる。最後に、予測された軌跡は、アクターの対応するアンカーポーズに従って、さらにグローバル座標に変換される。

G. Training

提案するSIMPLはエンドツーエンドで学習される。全体の損失関数は、回帰損失と分類損失の加重和である

$$\mathcal{L} = \omega\mathcal{L}_{reg} + (1 - \omega)\mathcal{L}_{cls},$$

ここで、$\omega \in [0, 1]$はこれらの成分のバランスをとるための重みであり、回帰タスクの重要性に対処するために$\omega = 0.8$とした。6]に従い、マルチモダリティを扱うためにWTA(winner-takes-all)戦略を用いる。各エージェントについて、最終的な変位誤差が最小となるものを選ぶことで、K個の仮説の中で最も予測される軌道k^*を求める。分類タスクに関しては、[6]と同様に、最大マージン損失を用いて、正モードと他モードを区別する。軌道回帰タスクでは、位置座標回帰に加えて、補助的な監視を提供するためにオプションのヨー角損失を導入し、その結果、以下のようになる。

$$\mathcal{L}_{reg} = \texttt{PosLoss}(\bar{Y}_{pos}, Y_{pos}^{k^*}) + \texttt{YawLoss}(\bar{Y}_{yaw}, Y_{yaw}^{k^*}),$$

ここで、$\triangle_{(-)}$はグランドトゥルース(GT)状態を表し、$Y_{(-)}^{k^*}$は勝者モードの予測位置とヨー角である。位置回帰損失として平滑 L1 損失を採用し、ヨー回帰損失を次のように指定する。

$$\texttt{YawLoss}(\bar{Y}_{yaw}, Y_{yaw}^{k^*}) = [1 - \texttt{CosSim}(\bar{Y}_{yaw}, Y_{yaw}^{k^*})]/2,$$

ここで、CosSim(-, -)は余弦類似度測定であり、2つの整列したヨーベクトルに対して1の値、2つの反対のヨーベクトルに対して-1の値が得られる。ヨー角損失を組み込むことで、連続する状態間の整合性が暗黙のうちに強化され、より滑らかで運動学的な実現可能性を持つ予測軌道ができ、特に低速エージェントの場合、より現実的な軌道が得られる。

IV. 実験結果

A. 実験セットアップ

1）データセット Argoverse1[22]とArgoverse2[23]の両方のモーション予測データセットで提案手法を評価する。Argoverse 1には、トレーニング用、検証用、テスト用のそれぞれ205942、39472、78143シーケンスが含まれる。各シーケンスは10Hzでサンプリングされ、タスクは2秒間の過去の観測(すなわち、H = 20, T = 30)に基づいて将来の3秒間の軌跡を予測することを含む。Argoverse 2の場合、トレーニング、検証、テスト用に200 000、25000、25000シーケンスで構成される。また、シーケンスは10Hzでサンプリングされ、与えられた履歴は5秒であり、将来の動作は6秒である(すなわち、H = 50, T = 60)。Argoverse 1とArgoverse 2の両方がHDマップを提供する。

2）メトリクス：主に、最小平均変位誤差(minADE_k)、最小最終変位誤差(minFDE_k)、ミス率(MR_k)、brier-minFDE_k など、マルチモーダル軌道予測でよく使われる標準的なメトリクスに従う。これらのメトリクスはすべて、K個の仮説の中で、単一のターゲットエージェントに対して、グランドトゥルースに対して最も予測された軌跡を評価する。minADE_kは予測軌跡とGTの平均ユークリッド距離であり、minFDE_kは終点での誤差のみを考慮する。MRは、得られたminFDE_kが2mを超えるシーケンスの割合である。Brier-minFDE_k はminFDE_k にさらにブリアスコア $(1 - p)^2$ を加える。詳細な定義については、[22, 23]を参照されたい。

3）実装の詳細：すべての潜在ベクトルに対して次元D = 128を設定し、対称グローバル特徴融合のために4つのSFT層と8つの注意ヘッドを積み重ねる。マルチモーダルデコーダでは、一般的な設定に従ってモード数K = 6を設定する。ベジエ曲線の次数nは、予測地平が異なるため、Argoverse 1では5、Argoverse 2では7に設定される。SIMPLは、8つのNvidia RTX 3090 GPUを搭載したサーバー上で、バッチサイズ128、50エポックを使用してエンドツーエンドで学習される。Adamオプティマイザを採用し、学習率を初期に1e-3に設定し、40エポック後に徐々に1e-4まで減少させる。

B. Results

1）最先端技術との比較：SIMPLと他の最先端手法を2つの大規模手法で比較する：Argoverse 1モーション予測データセットのテスト分割の結果。

上段と下段は、単一モデルとアンサンブル手法の結果である。最も良い結果は太字で、2番目に良い結果には下線が引かれている。b-minFDE 6 は公式ランキング指標である。＃は非公式実装のモデルサイズを表す。

Method	minADE_6	minFDE_6	MR_6	b-minFDE_6	#Param
LaneGCN [6]	0.870	1.362	16.2	2.053	3.7M
mmTrans [7]	0.844	1.338	15.4	2.033	2.6M
SceneTrans [8]	0.803	1.232	12.6	1.887	15.3M
HiVT [9]	**0.774**	**1.169**	12.7	1.842	2.5M
MacFormer [12]	0.819	1.216	**12.1**	1.827	2.4M
SIMPL (w/o ens)	0.793	1.179	12.3	**1.809**	**1.8M**
MultiPath++ [30]	0.790	1.214	13.2	1.793	21.1M♯
MacFormer [12]	0.812	1.214	12.7	1.767	2.4M
HeteroGCN [13]	0.789	1.160	11.7	1.751	-
Wayformer [36]	**0.768**	1.162	11.9	**1.741**	11.2M♯
SIMPL (w/ ens)	0.769	**1.154**	**11.6**	1.746	**1.8M**

表 II: 対称シーンモデリングに基づく手法の Argoverse 2 テスト分割の結果。結果は単一モデル(w/oアンサンブル)によるものである。最も良い結果は太字で、2番目に良い結果は下線で示されている。b-minFDE 6 は公式ランキング指標である。

Method	minADE_6	minFDE_6	MR_6	b-minFDE_6	#Param
HDGT [19]	0.84	1.60	21.0	2.24	12.1M
GoRela [20]	0.76	1.48	22.0	2.01	-
QCNet [21]	**0.65**	**1.29**	**16.0**	**1.91**	7.3M
SIMPL (w/o ens)	0.72	1.43	19.2	2.05	**1.9M**

モーション予測ベンチマーク。表 I は Argoverse 1 のテスト分割の定量的な結果である。上段は単一モデルの結果を示し、下段はアンサンブル技術を用いた手法の性能を示す。このようなシンプルな設計により、SIMPLはリストアップされたすべての手法の中で高い競争力のある結果を達成した。LaneGCN [6]、mmTransformer [7]、MacFormer [12]はエージェント中心の表現を利用しており、効率的なオンライン推論を妨げている。Scene Transformer [8]はシーン中心の表現を採用し、シングルパスのマルチエージェントの動き予測を可能にする。しかし、モデルサイズが大きく、性能が劣るため、データ量が多く、一般化しにくいことがわかる。HiVT[9]は、視点移動に対するロバスト性のために、特徴量融合時に相対的なポーズを明示的に考慮するが、SIMPLは、より良い性能を達成しながら、よりシンプルで軽い設計をもたらす。また、公正な比較のために、アンサンブルによる評価結果も報告する。k-meansクラスタリングに基づく8つのモデルのアンサンブルの後、SIMPLはMultiPath++ [30]のような強力なベースラインを上回り、Wayformer [36]のような最先端の手法に、はるかに少ないパラメータで競争力を持つ。Argoverse 2動作予測ベンチマークの評価結果をTab. II.SIMPLを、対称的なシーンモデリング技術を採用した他の最先端手法と比較する。SIMPLは、そのミニマリストアーキテクチャと驚くほどコンパクトなモデルサイズを特徴とし、競争力のある軌道予測結果を達成し、さらなる拡張と応用に有望である。

2) 推論待ち時間: 推論待ち時間の評価結果を図6に示す。すべての実験は、オリジナルのPyTorch実装を搭載したRTX 3060Ti GPUで実施した。まず、LaneGCN [6]とHiVT [9]と計算効率を比較する。エージェント中心として

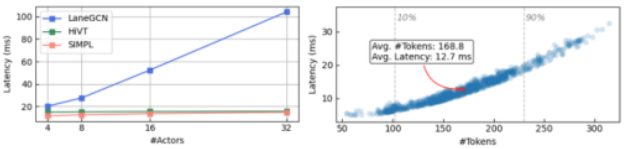

図6: Argoverse 1データセットにおける推論待ち時間の評価結果。左:ターゲットエージェントの数に対する各手法の平均推論レイテンシ。HiVTとSIMPLはマルチエージェントの動き予測においてリアルタイム性能を達成するが、エージェント中心のアプローチはスケールアップが難しい。右図: 右:実行時間とシーン内の総インスタンストークン数の関係。各ポイントは走行シーンを表し、そのすべてをリアルタイムで処理できる。

ベースライン、LaneGCNは各ターゲットの状態に対してシーンを正規化し、コンテキストをバッチ形式に整理する。HiVTと我々のSIMPLは共に共有コンテキストエンコーディングを採用し、1回のフォワードパスでマルチエージェント予測を行う。コンパクトな設計の恩恵により、SIMPLはHiVTよりもリアルタイム性能とわずかに優れた推論速度を達成した。また、対称シーンモデリングに基づく手法は、従来のエージェント中心の手法よりもはるかに効率的なマルチエージェント予測を可能にする。図6の右側は、完全な検証セットからランダムにサンプリングされた1,000シーンの待ち時間分布である。量子化のような高速化技術がなくても、SIMPLは高い推論速度を達成し、さらなる最適化後の実世界での搭載に有望である。

3) 定性的結果 Argoverse1と2の両データセットにおける定性的な結果を図7に示す。我々のSIMPLは、シーン内の複数のエージェントに対して、現実的で合理的かつ正確なマルチモーダルな将来の軌跡を同時に予測することができる。また、モーション予測データセットで学習したモデルに基づき、微調整(ゼロショット転送)を行わずにリアルタイム連続軌跡予測を行った場合の定性的な結果をArgoverseトラッキングデータセットで実証する。スナップショットは図1に描かれており、詳細な結果については添付の補足ビデオを参照されたい。

C. Ablation Study

1) 特徴融合モジュールについて: まず、提案するSFT層の設計を検討する。表IIIに示すように IIIに示すように、埋め込みサイズとSFT層数(M1→M3)の増加に伴い、SIMPLは全てのメトリクスでより良い性能を達成している。しかし、埋め込みサイズが128であることを考えると、SFT層の数を4層から6層(M4→M5)に増やすと、予測精度はわずかに向上するが、その代償として22%多くのパラメータを組み込むことになり、リアルタイムアプリケーションではあまり好まれない。また、各層のコンテキスト配列を用いて相対位置埋め込み(RPE)を更新することで、全体的な性能を大幅に向上させることができることがわかります(M3→M4)。これは、RPEの更新がエッジ特徴にノード特徴を組み込むことを含み、異なる意味インスタンス間の関係を学習するのに役立つためであると推測される。

2) 軌道パラメータ化について: さらに、異なる軌道パラメータ化手法の影響をTab. IV.30]で述べられた結論と同様に、単項基底多項式表現は、生の座標と比較して、大幅な性能低下をもたらす。一方、ベジエ曲線に基づく手法は、変位に関連するメトリクスにおいて、同レベルの結果を達成している。

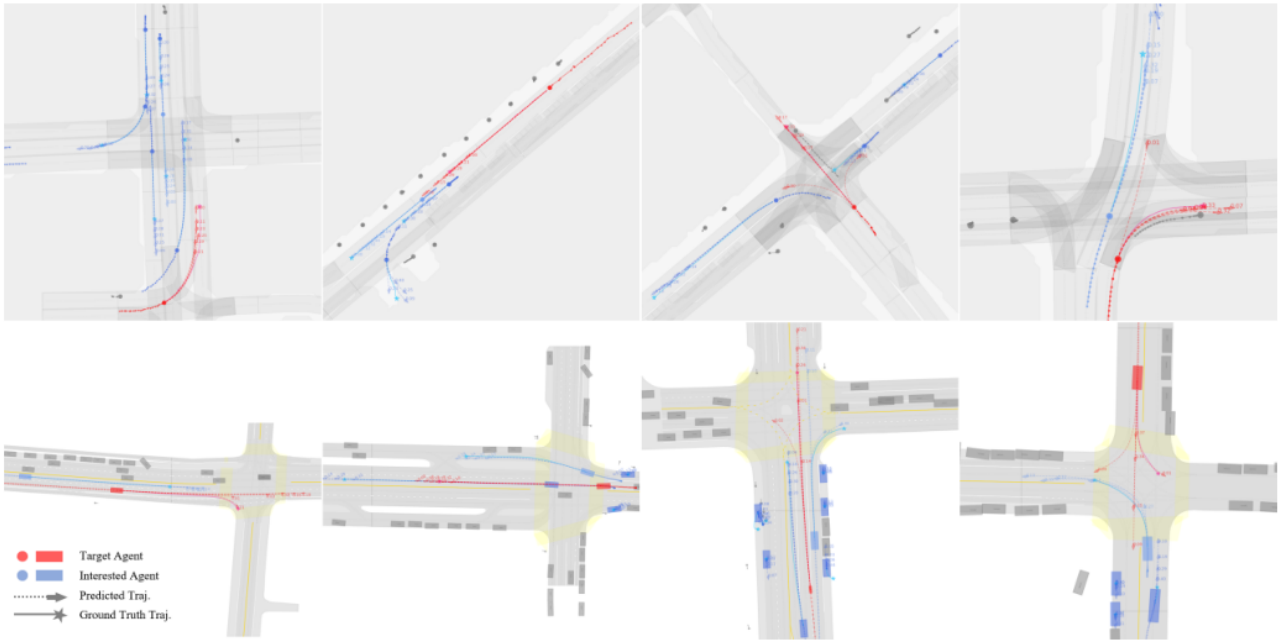

図1: Argoverse 1 (上)と Argoverse 2 (下)運動予測データセットにおける定性的結果。ターゲットエージェントは赤で、他の興味のあるエージェントは青で示されている。なお、シーン内の全エージェントの将来の動作は生成されるが、簡潔にするため、無視したエージェントの結果(灰色)は省略する。グランドトゥルースの終点は星印で、予測された軌跡は破線の曲線で描かれ、最終的なポーズは矢印で示されている。SIMPLは、複雑なシナリオにおいて、運転コンテキストの特徴を効果的に抽出し、特定のシーン制約に準拠した複数のエージェント軌道を生成する。(詳しくは拡大してください。)

TABLE III: Ablative study of the feature fusion module design on the Argoverse 1 validation split.

Model	Emb. Size	# Layers	RPE Upd.	minFDE_6	MR_6	b-minFDE_6
$\mathcal{M}1$	64	2	✗	1.237	12.8	1.848
$\mathcal{M}2$	64	4	✗	1.037	9.5	1.658
$\mathcal{M}3$	128	4	✗	0.993	9.0	1.607
$\mathcal{M}4$	128	4	✓	0.947	**8.1**	1.559
$\mathcal{M}5$	128	6	✓	**0.944**	8.4	**1.558**

表 IV: Argoverse 2 検証セットにおける軌道パラメータ化手法とヨー角損失のアブレーションスタディ。

Parameterization	Yaw loss	minADE_6	minFDE_6	minAYE_6	minFYE_6
Raw coords	✗	**0.780**	**1.452**	0.134	0.151
Polynomial	✗	0.861	1.738	0.146	0.278
Bézier curve	✗	**0.780**	1.457	0.137	0.297
Bézier curve	✓	0.783	**1.452**	**0.055**	**0.076**

ベジエ曲線の係数が特定の空間的意味を持つ制御点であるのに対して、単項式の数値的不均衡による性能低下は、座標を直接回帰することによる難易度の大きな差はない。異なるパラメタリゼーション手法の予測係数分布を比較すると(図8参照)、ベジエ曲線の分布が単項基底よりも規則的であり、このタスクを容易にする可能性があることがわかる。

3) 補助損失関数について: 連続表現を活用することで、物理的な制約に違反することなく、高次の物理量にアクセスできるようにすることもできる。したがって、ネットワーク・アーキテクチャを変更することなく、方位角のような量に対する損失関数を自然に導入することができる。III-Gで導入したヨーロスを評価するために、最小平均ヨー誤差(minAYE_k)と最小最終ヨー誤差(minFYE_k)を導入し、ラジアン単位の絶対角度差を直接計算する。表 IVから、以下のことがわかる。IVから、以下のことがわかる。

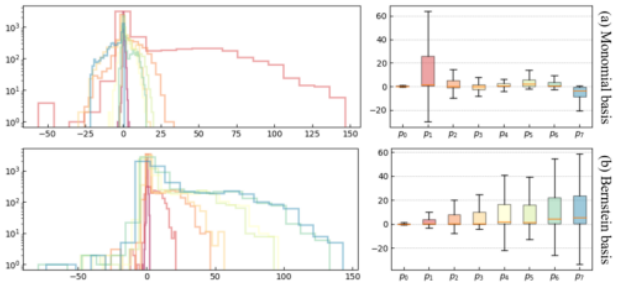

図8:単項基底多項式(a)とBernstein基底多項式の予測x軸係数の分布(b)。左の列は係数分布のヒストグラム、右の列は対応するボックスプロットを示す。異なる次数の係数の分布は、異なる色で示されている。単項基底多項式の1次係数は他のものよりはるかに広いスパンを持つが、ベジエ曲線の係数(すなわち制御点)は空間的な意味からより規則的であることが分かる。

ヨーロスはヨー角の精度を大幅に向上させ、実世界のアプリケーションに非常に有利である。

D. Extensibility

以上のように、我々のSIMPLは最も単純なネットワークアーキテクチャ設計に従っており、さらなる拡張のためのスペースを残している。この利点を実証するために、最近の最先端アプローチ[11, 21, 37]で広く採用されている、反復提案洗練の考え方に従った、追加の単純な軌道デコーダを追加する。我々はバニラSIMPLに修正を加えず、予測されたマルチモーダル軌道を初期提案とみなす。簡単のため、ここではターゲットエージェントの予測軌道のみを改良する。11,37]と同様に、各提案軌道に対して、まず特徴ベクトルに再符号化する。次に、各提案の近傍のインスタンス特徴をある範囲内で収集し、標準的なTransformerデコーダに基づく特徴融合モジュールが続き、

対応するTABLE Vにローカルコンテキストをアテンションする: Argoverse 1の検証およびテスト分割における拡張性実験の定量的結果。

Split	Method	minADE_6	minFDE_6	MR_6	b-minFDE_6
Val	SIMPL	0.658	0.947	**8.1**	1.559
	SIMPL-R	**0.651**	**0.946**	8.2	**1.542**
Test	SIMPL	0.793	1.179	12.3	1.809
	SIMPL-R	**0.783**	**1.173**	**12.1**	**1.781**

提案特徴量。精密化後、提案特徴量は別の単純なMLPベースのデコーダに送られ、最終的な予測軌跡とその確率スコアを得る。学習過程はバニラSIMPLと同じであり、すなわち、最もよく予測された提案を正の軌跡とするWTA戦略を採用する。拡張モデルをSIMPL-Rと表記し、その結果をTab. V. このような単純なプラグアンドプレイのポストリファインメントモジュールにより、SIMPL-Rは全体的に優れた性能を達成し、コンパクトなアーキテクチャがスケーラブルで、様々な異なるタスクのバックボーンとして使用できることが期待できることを示している。また、SIMPLは自己教師付き学習[38, 39]などの他の最近の技術とスムーズに統合できることに注意する。もう一つの今後の課題として残す。

V. Conclusion

本論文では、自律走行のためのシンプルで効率的なマルチエージェント動作予測ベースラインを提示する。提案する対称融合Transformerを活用することで、効率的な大域的特徴融合を実現し、視点移動に対する頑健性を保持する。Bernstein基底多項式に基づく連続軌跡のパラメタリゼーションは、下流モジュールとの高い互換性を提供する。大規模な公開データセットを用いた実験結果から、SIMPLは他の最先端手法と同レベルの精度を得ることができ、モデルサイズと推論速度の点でより有利であることが示された。

References

[1] H. Cui, *et al.*, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *ICRA*. IEEE, 2019, pp. 2090–2096.

[2] T. Zhao, *et al.*, "Multi-agent tensor fusion for contextual trajectory prediction," in *CVPR*, 2019, pp. 12 126–12 134.

[3] Y. Chai, *et al.*, "MultiPath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," *arXiv preprint arXiv:1910.05449*, 2019.

[4] T. Phan-Minh, *et al.*, "CoverNet: Multimodal behavior prediction using trajectory sets," in *CVPR*, 2020, pp. 14 074–14 083.

[5] J. Gao, *et al.*, "VectorNet: Encoding HD maps and agent dynamics from vectorized representation," in *CVPR*, 2020, pp. 11 525–11 533.

[6] M. Liang, *et al.*, "Learning lane graph representations for motion forecasting," in *ECCV*. Springer, 2020, pp. 541–556.

[7] Y. Liu, *et al.*, "Multimodal motion prediction with stacked transformers," in *CVPR*, 2021, pp. 7577–7586.

[8] J. Ngiam, *et al.*, "Scene Transformer: A unified architecture for predicting multiple agent trajectories," *arXiv preprint arXiv:2106.08417*, 2021.

[9] Z. Zhou, *et al.*, "HiVT: Hierarchical vector transformer for multi-agent motion prediction," in *CVPR*, 2022, pp. 8823–8833.

[10] L. Zhang, *et al.*, "Trajectory prediction with graph-based dual-scale context fusion," in *IROS*. IEEE, 2022, pp. 11 374–11 381.

[11] S. Shi, *et al.*, "Motion transformer with global intention localization and local movement refinement," *NeurIPS*, vol. 35, pp. 6531–6543, 2022.

[12] C. Feng, *et al.*, "Macformer: Map-agent coupled transformer for real-time and robust trajectory prediction," *IEEE Robot. Autom. Lett.*, 2023.

[13] X. Gao, *et al.*, "Dynamic scenario representation learning for motion forecasting with heterogeneous graph convolutional recurrent networks," *IEEE Robot. Autom. Lett.*, vol. 8, no. 5, pp. 2946–2953, 2023.

[14] S. Casas, *et al.*, "Implicit latent variable model for scene-consistent motion forecasting," in *ECCV*. Springer, 2020, pp. 624–641.

[15] M. Werling, *et al.*, "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," *Intl. J. Robot. Res.*, vol. 31, no. 3, pp. 346–359, 2012.

[16] W. Ding, *et al.*, "Safe trajectory generation for complex urban environments using spatio-temporal semantic corridor," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2997–3004, 2019.

[17] T. Buhet, *et al.*, "PLOP: Probabilistic polynomial objects trajectory planning for autonomous driving," *arXiv preprint arXiv:2003.08744*, 2020.

[18] X. Jia, *et al.*, "Towards capturing the temporal dynamics for trajectory prediction: a coarse-to-fine approach," in *CoRL*. PMLR, 2023, pp. 910–920.

[19] ——, "HDGT: Heterogeneous driving graph transformer for multi-agent trajectory prediction via scene encoding," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023.

[20] A. Cui, *et al.*, "GoRela: Go relative for viewpoint-invariant motion forecasting," in *ICRA*. IEEE, 2023, pp. 7801–7807.

[21] Z. Zhou, *et al.*, "Query-centric trajectory prediction," in *CVPR*, 2023, pp. 17 863–17 873.

[22] M.-F. Chang, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *CVPR*, 2019, pp. 8748–8757.

[23] B. Wilson, *et al.*, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," *arXiv preprint arXiv:2301.00493*, 2023.

[24] A. Alahi, *et al.*, "Social LSTM: Human trajectory prediction in crowded spaces," in *CVPR*, 2016, pp. 961–971.

[25] A. Vemula, *et al.*, "Social attention: Modeling attention in human crowds," in *ICRA*. IEEE, 2018, pp. 4601–4607.

[26] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[27] P. Veličković, *et al.*, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[28] A. Vaswani, *et al.*, "Attention is all you need," *NeurIPS*, vol. 30, 2017.

[29] H. Cui, *et al.*, "Deep kinematic models for kinematically feasible vehicle trajectory predictions," in *ICRA*. IEEE, 2020, pp. 10 563–10 569.

[30] B. Varadarajan, *et al.*, "MultiPath++: Efficient information fusion and trajectory aggregation for behavior prediction," in *ICRA*. IEEE, 2022, pp. 7814–7821.

[31] F. Gao, *et al.*, "Online safe trajectory generation for quadrotors using fast marching method and Bernstein basis polynomial," in *ICRA*. IEEE, 2018, pp. 344–351.

[32] S. Deolasee, *et al.*, "Spatio-temporal motion planning for autonomous vehicles with trapezoidal prism corridors and bézier curves," in *ACC*. IEEE, 2023, pp. 3207–3214.

[33] R. Girgis, *et al.*, "Latent variable sequential set transformers for joint multi-agent motion prediction," *arXiv preprint arXiv:2104.00563*, 2021.

[34] C. R. Qi, *et al.*, "PointNet: Deep learning on point sets for 3d classification and segmentation," in *CVPR*, 2017, pp. 652–660.

[35] S. Shi, *et al.*, "MTR++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying," *arXiv preprint arXiv:2306.17770*, 2023.

[36] N. Nayakanti, *et al.*, "Wayformer: Motion forecasting via simple & efficient attention networks," in *ICRA*. IEEE, 2023, pp. 2980–2987.

[37] S. Choi, *et al.*, "R-Pred: Two-stage motion prediction via tube-query attention-based trajectory refinement," *arXiv preprint arXiv:2211.08609*, 2022.

[38] P. Bhattacharyya, *et al.*, "SSL-Lanes: Self-supervised learning for motion forecasting in autonomous driving," in *CoRL*. PMLR, 2023, pp. 1793–1805.

[39] J. Cheng, *et al.*, "Forecast-MAE: Self-supervised pre-training for motion forecasting with masked autoencoders," *arXiv preprint arXiv:2308.09882*, 2023.