

ASCENT: Transformer-Based Aircraft Trajectory Prediction in Non-Towered Terminal Airspace

Alexander Prutsch¹, David Schinagl¹ and Horst Possegger¹

Abstract—Accurate trajectory prediction can improve General Aviation safety in non-towered terminal airspace, where high traffic density increases accident risk. We present ASCENT, a lightweight transformer-based model for multi-modal 3D aircraft trajectory forecasting, which integrates domain-aware 3D coordinate normalization and parameterized predictions. ASCENT employs a transformer-based motion encoder and a query-based decoder, enabling the generation of diverse maneuver hypotheses with low latency. Experiments on the TrajAir and TartanAviation datasets demonstrate that our model outperforms prior baselines, as the encoder effectively captures motion dynamics and the decoder aligns with structured aircraft traffic patterns. Furthermore, ablation studies confirm the contributions of the decoder design, coordinate-frame modeling, and parameterized outputs. These results establish ASCENT as an effective approach for real-time aircraft trajectory prediction in non-towered terminal airspace.

I. INTRODUCTION

Accurate trajectory predictions are crucial for improving airspace safety systems, especially for General Aviation (GA) flights near non-towered airports. GA includes all non-commercial civilian flights, such as recreational flying, business aviation, and aerial work (*e.g.*, agricultural aviation). It represents the majority of the aircraft fleet; for example, around 90% of all aircraft registered in the United States fall under this category [1]. Commercial Aviation (CA), which focuses on the transport of passengers or cargo, operates under stringent safety standards: flights operate in controlled airspace according to structured procedures, with departures and arrivals conducted at towered airports under the active management of air traffic control (ATC). However, most airports are *non-towered*; it is estimated that only about 4% of the roughly 20,000 airports in the US have control towers [2]. With no ATC active, pilots must rely on standard procedures, for example Federal Aviation Administration (FAA) *Standard Airport Traffic Patterns* [3], and their own judgment to maintain safety. GA flights frequently operate between non-towered airports, thereby facing a *higher risk of accidents*. The accident rate in GA is approximately 30 times higher than in commercial aviation (CA) [4], with fatal accidents occurring predominantly in the vicinity of airports.

The wide spread of GA, together with its relatively high accident rate, highlights the need for safety systems in non-towered airspace. One promising approach is the prediction of future aircraft trajectories, which typically relies on historical aircraft movement data combined with global context in-

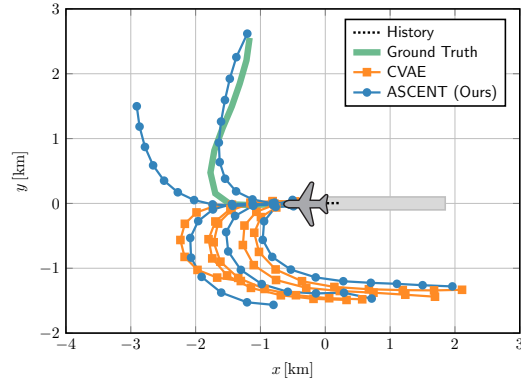


Fig. 1. Comparison between a CVAE-based decoder, widely applied in aircraft trajectory prediction, and our ASCENT model employing a query-based decoder. The scenario shows an aircraft take-off: our model, leveraging learnable mode queries, predicts a diverse set of trajectories that also capture the actual aircraft movement, while the baseline only predicts the common traffic pattern at the data-collection airport.

formation. The predictions can help detect potential conflicts at an early stage [5], thereby improving GA safety in non-towered terminal areas. Additionally, trajectory prediction can be applied in other tasks, *e.g.*, to enhance aircraft traffic flow [6] and improve arrival time/delay prediction [7]. Previous research on aircraft trajectory prediction has introduced datasets from non-towered airspace [8], [9] and developed deep learning-based methods [8], [10]–[13].

Analogously, trajectory prediction has become a key challenge in autonomous driving, where anticipating the future positions of surrounding traffic participants strongly influences self-driving vehicle decision-making and path planning. Interest in autonomous driving has therefore driven a significant research focus on motion forecasting for self-driving vehicles [14]–[22]. While methods that originate from the autonomous driving domain target a similar task, they *cannot be directly applied to aircraft trajectory prediction* due to *distinct challenges*: unlike road traffic, airplanes do not follow strict lane networks; only runways and airspace procedures define how aircraft approach and depart airports. Moreover, aircraft operate in three-dimensional space, whereas bird’s-eye-view prediction is mostly sufficient for vehicles. Furthermore, trajectory prediction benchmarks for aviation [8], [9] feature significantly longer scenarios and prediction horizons spanning several minutes compared to only a few seconds in the automotive domain.

Although recent advances have been made, aircraft trajectory prediction remains less extensively studied than its counterpart in autonomous driving. In the self-driving domain, state-of-the-art methods typically employ transformer-

¹ Corresponding author: alexander.prutsch@tugraz.at. All authors are with the Institute of Visual Computing, Graz University of Technology, Austria. This work was partially funded by the Austrian Research Promotion Agency (FFG) under the project SAFER (894164).

based [23] modules to encode historical motion data, and combine it with structured environmental information, *e.g.*, on lane topology [24]. Recent advances have been largely driven by improved coordinate modelings [17], decoder designs [21], [25] and refined processing schemes [20]–[22], [25]. At the same time, the field of aviation trajectory prediction [12], [13] follows different approaches, particularly in terms of decoder architectures, highlighting a divergence between the two domains. To advance trajectory prediction for General Aviation, we address the unique challenges of aircraft motion and propose a novel approach leveraging insights from state-of-the-art methods in the self-driving domain. We propose an *Aircraft Sequence Encoding Transformer* (ASCENT) featuring a lightweight yet highly capable transformer-based architecture. In the decoder design, we diverge from common practices in aircraft trajectory prediction by using a mode query-based approach [19], which better suits the problem domain compared to diffusion or variational autoencoder-based modules of related work [11]–[13]. As highlighted in Figure 1, our approach generates more diverse outputs, thereby better covering the range of possible future actions. We evaluate our approach on the widely used TrajAir dataset [8] and provide results on the recently proposed TartanAviation dataset [9].

In summary, our main contributions include:

- We propose a novel architecture for aircraft trajectory prediction, achieving a new state-of-the-art on the TrajAir dataset across multiple evaluation settings.
- To the best of our knowledge, we are the first to report trajectory prediction results on the TartanAviation dataset, including cross-dataset evaluations.
- We perform an extensive ablation study of our model demonstrating the effectiveness of our architecture.

II. RELATED WORK

A. Aircraft Trajectory Prediction

Patrikar *et al.* [8] propose alongside their TrajAir dataset also a baseline method (TrajAirNet) for aircraft trajectory prediction in non-towered airspace. Initially, the sequence of 3D positions of each agent, given in absolute coordinates, are encoded using temporal convolutional networks [26], before interactions between all agents are modeled using graph attention networks [27]. The airplane trajectories are decoded using conditional variational autoencoders [28]. To model the flight dynamics, the model predicts acceleration values which are then integrated to 3D positions using Verlet integration [29]. Social-PatteRNN [10] proposes a variational RNN (VRNN)-based [30] approach with a novel context modeling which is also evaluated on the TrajAir dataset.

Also Yin *et al.* [11] focus on context modeling for trajectory prediction and propose a diffusion-based approach. Evaluation is conducted on a custom, non-public dataset recorded at the towered Singapore Changi Airport. Their model combines a Trajectron++-based [31] motion encoder that operates on absolute coordinates with a separate context encoder. The latter processes additional features such as

landing time and target runway, which are only applicable in towered airspace. The trajectory and context features are concatenated and future trajectories are predicted using a decoder inspired by MID [32], which models a reverse diffusion process. The follow-up work ATP [12] leverages a local history of similar past aircraft behaviors, which achieves strong results on TrajAir as well as the private Singapore Changi Airport dataset. Recently, GooDFlight [13] also proposes a diffusion-based model for flight trajectory prediction. This approach adopts goal-oriented modeling: it first predicts goal coordinates, which are then used in a goal-guided decoder. GooDFlight currently achieves state-of-the-art results on the TrajAir dataset.

Despite extensive research on aircraft trajectory prediction, existing models differ substantially from state-of-the-art in autonomous driving. Current approaches in aviation often employ decoders based on generative stochastic modeling, *i.e.*, variational methods [8], [10] or diffusion processes [11]–[13], to predict multi-modal trajectories. Generative approaches have shown particular strength in domains like pedestrian trajectory prediction [33]–[35], where agents move freely and unpredictably. However, generative designs are not adopted in leading self-driving models [17], [20], [21], [25]. Although GA aircraft do not strictly follow lanes as cars do, they still adhere to predefined traffic patterns, resulting in structured and discrete motion modes. Given the maturity and demonstrated accuracy of trajectory prediction in autonomous driving, it is natural to ask whether insights and architectures from that field can be adapted to the aviation domain. Such cross-domain transfer offers a promising direction for advancing aircraft trajectory prediction, despite the inherent differences between the two problem settings.

B. Trajectory Prediction for Autonomous Driving

Trajectory prediction for self-driving vehicles is a well-studied task. Large-scale benchmark datasets like the Waymo Open Motion Dataset [36] and nuScenes [37] have significantly driven advances in motion prediction modeling. In contrast to airspace trajectory prediction, vehicle movements are heavily restricted by the lane topology and other environmental context like traffic lights and lane markings. In addition, methods mostly operate only in a 2D bird’s eye view representation. Motion prediction methods for autonomous driving commonly follow the same high-level architecture, where first historical agent movements and map elements are encoded in a local coordinate system, before global positional embeddings are employed and the relationships between scene elements are learned. After initial works explored CNNs [38] or LSTMs [39], attention blocks [23] have emerged as a popular approach for encoding agent motion behavior [14], [19]. Scene encoding is commonly done using graph neural networks [17], [24], [40]–[42] or transformers [14], [15], [19], [43]. Global positional embeddings can be implemented using MLP-based [19], sinusoidal [14], or Fourier-space embeddings [17]. To decode multi-modal trajectories, the common practice is to use learnable mode queries which model different potential

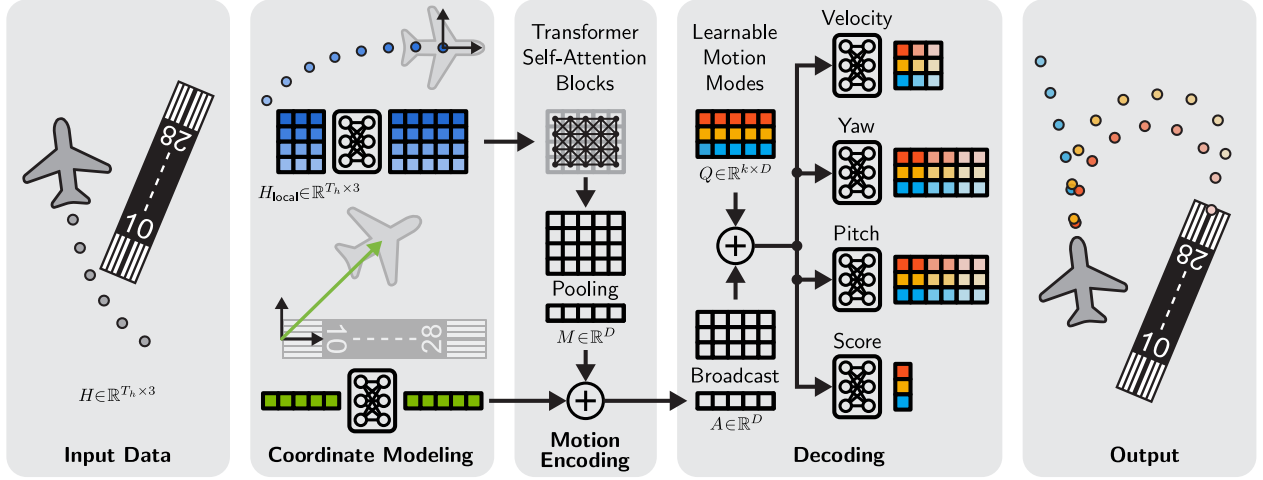


Fig. 2. Overview of our ASCENT architecture. For clarity of presentation, visualizations are reduced to 2D. First, we normalize the historical 3D flight path with respect to the current aircraft pose. Next, we encode this local history using attention blocks and combine it with global context through positional embeddings. To generate multi-modal future trajectories, we employ learnable mode queries that capture potential actions, such as landing or turning.

future agent directions. The mode queries can either be used in simple MLP-based decoders [19], [44] or as queries in a detection transformer (DETR)-like decoder [14], [20].

Methods developed for autonomous driving cannot be directly applied to aircraft trajectory prediction, as they rely heavily on lane topologies for guidance and are restricted to a 2D space. They are also tailored to much denser traffic scenarios but typically operate only over short time horizons of a few seconds – insufficient for aviation needs. In addition, these models are often large and require extensive, diverse training datasets. In contrast to terminal airspace, where the environment is static, methods for self-driving vehicles must generalize across diverse and dynamically evolving road networks. Nevertheless, concepts from autonomous driving like local motion encoding with global positional embeddings and mode query-based decoders represent promising directions for advancing aircraft trajectory prediction.

III. AIRCRAFT SEQUENCE ENCODING TRANSFORMER

In this section, we formally define the aircraft trajectory prediction problem and introduce our novel ASCENT architecture (see Figure 2 for an overview), which integrates best practices from autonomous driving research and extends them to the unique challenges of 3D flight paths. We first describe our input coordinate modeling, where positional and angular normalization are applied to effectively capture motion patterns from historical trajectories. To preserve global context after normalization, we introduce a novel 3D positional embedding. Both normalization and embedding must account for the horizontal (yaw) and vertical (pitch) orientation of aircraft, unlike vehicles that typically vary only in yaw. We then outline our motion pattern encoding. Finally, we detail our decoder design, where learnable mode queries and flight parameter outputs are employed to better align the architecture with the characteristics of the aviation domain.

A. Problem Statement

Aircraft trajectory prediction is the task of forecasting the 3D positions of an aircraft over T_f future steps, denoted as

$F_{GT} \in \mathbb{R}^{T_f \times 3}$, given its historical motion over T_h steps ($H \in \mathbb{R}^{T_h \times 3}$). Since future motion can correspond to different maneuvers, *e.g.*, landing or turning, models typically predict a set of k trajectory candidates, resulting in multi-modal outputs $F \in \mathbb{R}^{k \times T_f \times 3}$ with associated probability scores $S \in \mathbb{R}^k$. In non-towered terminal airspace, the historical data H is typically referenced to a static Automatic Dependent Surveillance–Broadcast (ADS-B) receiver. Additionally, auxiliary context such as weather conditions or the trajectories of neighboring aircraft can be incorporated.

B. Model Architecture

1) *Coordinate Modeling*: Generally, the future path of an aircraft depends on its historical motion pattern and the global position context, *e.g.*, distance and orientation to the runway. Aircraft trajectory prediction datasets [8], [9] are commonly recorded using a static scene coordinate system. By directly using the global coordinates it is possible to model both, however, this can lead to a global position bias and hinders effectively learning local motion patterns, such as turns, dives or climb flights. To circumvent this, it is a common practice to normalize each agent path with respect to the current agent position to obtain an agent-centric local coordinate system. Naively, this can be either done by just shifting the trajectory by the object center, or more sophisticated by also aligning the orientation of the agent with the coordinate axis *e.g.*, aircraft looking in positive y-direction. Figure 3 shows a comparison of global and normalized coordinates using example data from the TrajAir dataset. In our approach, we apply positional and angular normalization to obtain H_{local} which can be used to effectively learn motion patterns. We estimate the orientation of the aircraft using the last and second-to-last position step. Following, we compute the horizontal (yaw angle γ) and the vertical (pitch angle θ) orientation using the arctangent. Normalization is then done by subtracting the most-recent aircraft position and rotation around γ and θ .

Using only agent-centric local coordinates neglects global information, *e.g.*, the relationship between the aircraft pose

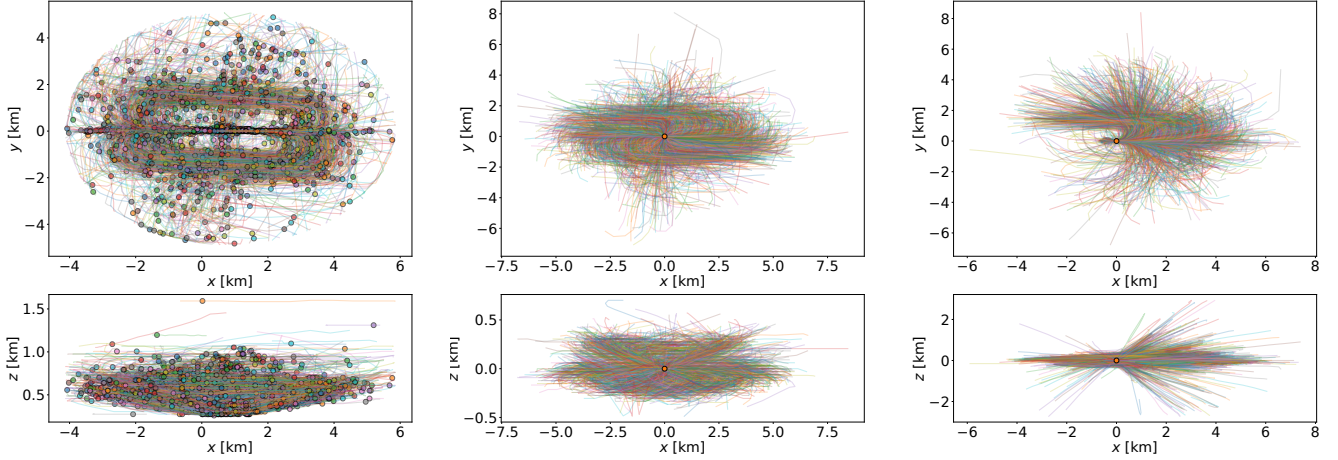


Fig. 3. Comparison of different coordinate system normalization approaches on trajectories from the TrajAir dataset [8]. We show the current aircraft position (circular markers), along with historical (11 s) and future (120 s) trajectories. We compare global coordinates (left), object-centered normalization (middle), and our combined positional-angular normalization (right). For each method, the bird’s-eye view is shown on top, with vertical trajectories below. The global coordinate visualization shows aircraft approaching and departing from the runway, which is aligned with the x-axis. Normalizing with respect to the aircraft position moves all centers to the origin, while angular normalization aligns orientations, producing structured clusters of similar maneuvers, such as left and right turns. The top-right plot highlights a higher frequency of left turns, reflecting the left-traffic pattern at the KBTP airport.

and a runway. A common practice in the automotive domain to preserve global context is the use of positional embeddings. We employ a novel 3D positional embedding using the 3D position (x, y, z) and the orientation angles $(\gamma$ and $\theta)$, which we compute during normalization.

2) *Motion Encoding*: After normalization to local coordinates, we encode the historical aircraft path $H_{\text{local}} \in \mathbb{R}^{T_h \times 3}$ using self-attention blocks [23]. Before applying the transformer blocks, we first project H_{local} to our feature space using a linear layer [18] and add temporal encodings: we encode a numerical time index to a temporal embedding $\mathbb{R}^D$ using a linear layer. After self-attention we apply max-pooling to reduce the dimensionality [18]. As a result we obtain a single encoded motion feature vector $M \in \mathbb{R}^D$ for an agent. To model the global positional context, we then encode the pose $(x, y, z, \gamma, \theta)$ using an MLP and add it to M to get our feature vector $A \in \mathbb{R}^D$.

3) *Decoding with Learnable Mode Queries*: ASCENT employs a simple multilayer perceptron (MLP)-based decoder with learnable queries. This design has been proven to be a simple yet effective method for generating multi-modal trajectory outputs in the automotive domain [19], [44]. We take the encoded agent feature A , broadcast it to $\mathbb{R}^{k \times D}$ and add k learnable mode queries $Q \in \mathbb{R}^{k \times D}$. Following, we decode probability scores $S \in \mathbb{R}^k$ using a MLP. To obtain the future positions $F \in \mathbb{R}^{k \times T_f \times 3}$ we do not directly predict the 3D positions, but rather predict flight parameters which we then use to compute the future positions. Using kinematics instead of direct positional outputs allows the model to more easily capture the motion patterns of real-world actors [49]. Our network predicts the yaw and pitch angles of the aircraft and the flying speed. The speed is predicted using a MLP with linear outputs. To model the angular components, our model first predicts the sine and cosine of the target angle, from which the angle is then computed via the arctangent. Using

this, we wrap the angles to $[-\pi, \pi]$ while also achieving a stable loss behavior. After predicting the individual components, we first compute the 3D positions based on the individual flight parameters. Following, we apply a coordinate transformation inverse to our normalization, to transfer the predictions back to the global 3D space.

IV. EXPERIMENTAL SETUP

A. Datasets and Data Preprocessing

We evaluate our method on two datasets, TrajAir [8] and TartanAviation [9], both of which contain 3D aircraft trajectory data recorded via Automatic Dependent Surveillance-Broadcast (ADS-B). The TrajAir dataset [8] was recorded at the non-towered Pittsburgh-Butler Regional Airport (ICAO: KBTP) across 111 days between September 2020 and April 2021. The TartanAviation [9] extends the ADS-B data from the TrajAir dataset with additional 222 days of recordings from the Pittsburgh-Butler Regional Airport. The dataset also includes data from a second airport, the towered Allegheny County Airport (ICAO: KAGC). It also contains additional data (images and ATC voice), which are not relevant for our trajectory forecasting approach.

Regarding the TrajAir dataset [8] we follow the official implementation¹ and utilize the preprocessed data provided by the authors. Their preprocessing pipeline removes corrupted and duplicated data points, filters out aircraft flying above 6000 feet MSL (mean sea level) or beyond a 5 kilometer radius away from the endpoints of the runway. It also excludes empty sequences where no aircraft is recorded. The framework provides training and validation sets for five distinct data splits: four splits each containing one week of data (*7Days-1-4*) and one comprehensive split containing all available data (*111Days*). Analogously, we also use the official implementation² and preprocessed data

¹<https://github.com/castacks/trajairnet>

²<https://github.com/castacks/TartanAviation/>

TABLE I

TRAJECTORY PREDICTION RESULTS ON THE DEDICATED TRAJAIR SPLITS, ALONG WITH AVERAGE RESULTS ACROSS ALL FOUR SPLITS. ALL MODELS PREDICT $K=5$ TRAJECTORIES. THE UPPER GROUP FOLLOWS THE EXPERIMENT SETUP OF [8] (USING 11 s INPUT). THE LOWER GROUP COMPARES TO RESULTS FROM [12], USING THEIR SETUP (16 s INPUT). MODELS ARE SORTED BY AVERAGE minFDE₅. ERRORS ARE REPORTED IN KM.

Input: 11 s Method	7Days-1		7Days-2		7Days-3		7Days-4		7Days-Avg	
	minADE ₅	minFDE ₅	minADE ₅	minFDE ₅	minADE ₅	minFDE ₅	minADE ₅	minFDE ₅	minADE ₅	minFDE ₅
Constant Velocity	1.79	4.08	1.90	4.31	1.92	4.30	1.82	4.16	1.86	4.21
TransformerTF [45]	1.58	3.85	1.69	4.10	1.97	4.36	1.79	4.19	1.76	4.13
STG-CNN [46]	1.19	2.35	1.36	2.70	1.33	2.67	1.17	2.29	1.26	2.50
Nearest Neighbor	3.13	2.70	1.92	1.99	3.41	2.69	2.59	2.58	2.76	2.49
TrajAirNet [8]	<u>0.73</u>	<u>1.42</u>	<u>0.81</u>	<u>1.63</u>	<u>0.86</u>	<u>1.72</u>	<u>0.71</u>	<u>1.41</u>	<u>0.78</u>	<u>1.55</u>
ASCENT (Ours)	0.33	0.57	0.37	0.62	0.36	0.59	0.33	0.56	0.35	0.58

Input: 16 s Method	7Days-1		7Days-2		7Days-3		7Days-4		7Days-Avg	
	minADE ₅	minFDE ₅	minADE ₅	minFDE ₅	minADE ₅	minFDE ₅	minADE ₅	minFDE ₅	minADE ₅	minFDE ₅
Constant Velocity	1.68	4.03	1.71	4.10	1.76	4.19	1.65	4.00	1.70	4.08
Nearest Neighbor	0.86	1.69	0.90	1.79	1.01	1.93	0.83	1.67	0.90	1.77
TrajAirNet [8]	0.70	1.37	0.75	1.48	0.84	1.62	0.69	1.40	0.75	1.47
MID [32]	0.61	1.13	0.61	1.12	0.63	1.18	0.58	1.11	0.61	1.13
Expert-Traj [47]	0.57	1.05	0.56	1.03	0.63	1.15	0.53	0.99	0.57	1.06
LBA [48]	<u>0.48</u>	0.88	0.51	0.91	0.56	1.04	0.46	0.86	0.50	0.92
ATP [12]	<u>0.48</u>	<u>0.85</u>	<u>0.50</u>	0.91	0.54	0.98	<u>0.45</u>	<u>0.82</u>	<u>0.49</u>	0.89
CoATP w/ Abs. Loc. [11]	0.50	0.88	<u>0.50</u>	<u>0.89</u>	<u>0.52</u>	<u>0.92</u>	0.47	0.83	0.50	<u>0.88</u>
ASCENT (Ours)	0.31	0.56	0.35	0.61	0.34	0.61	0.33	0.57	0.33	0.59

for the TartanAviation dataset [9]. Since no 5 km radius filtering is applied, the dataset covers a wider range from the KBTP airport compared to TrajAir. Since there are no official data splits for TartanAviation, we employ a custom split for each airport based on the processed sequence files: We take the first and last 25% of files as one split (named S1) and the middle 50% as a second split (S2). This reduces the impact of seasonal bias, with S1 covering potentially different seasonal conditions and S2 representing a more uniform set.

B. Implementation Details

We set the feature dimension D in our model to 128 and use two self-attention blocks in our encoder. In the decoder, each head (speed, yaw angle, pitch angle, and probability score) uses a 3-layer MLP with $2 \cdot D$ units in the hidden layers. The output layers are set to T_f for the velocity head and to $2 \cdot T_f$ in the angular heads, where T_f is the number of future steps. The output layer for the probability head is a single value per trajectory. The positional embeddings are implemented using a shallow MLP including two layers with dimension D . We model the positional embedding of each actor based on its position x, y, z , yaw γ , and pitch θ as $(x, y, z, \sin \gamma, \cos \gamma, \sin \theta, \cos \theta)$ to wrap angles.

We train our model on a single NVIDIA V100 GPU for 20 epochs using a batch size of 32 and an initial learning rate of 0.001, which is halved after 10 and 15 epochs. Loss computation is done using a winner-takes-all strategy: only the candidate trajectory that best fits the ground truth is considered for optimization (lowest L_2 distance). We employ a smooth L1 loss [50] as regression loss and use a standard cross-entropy loss as classification loss, *i.e.*, the best-fitting trajectory is assigned the highest probability score.

C. Metrics

We evaluate using the standard aircraft trajectory prediction metrics [8], [13]: the average displacement error (ADE_k) and final displacement error (FDE_k) reported in kilometers.

ADE is the mean displacement (L_2 distance) between prediction and ground truth trajectory across all future timesteps, whereas the FDE only takes the endpoints into account. Following the standard protocol, both metrics are reported using the best-fitting prediction out of the k candidates.

V. RESULTS AND DISCUSSION

We present our main results for trajectory prediction on the TrajAir [8] dataset in Tables I and II. Related work [8], [12], [13] report results using different historical data horizons as input and different numbers of output trajectory modes. To provide an extensive evaluation and ensure comparability to state-of-the-art methods, we provide results for our approach on all three experiment configurations. In Table I we compare our approach on the four dedicated one week data splits from the TrajAir dataset. We follow the standard setup from the dataset paper [8], using an 11 s history (sampled at 1 Hz) to predict the subsequent 120 s at 0.1 Hz, as well as the setting from related work [12], which uses a 16 s history and a 120 s prediction horizon, both sampled at 0.2 Hz. Our method significantly outperforms related work in both settings across all data splits. Comparing both experimental settings shows that the advantage of using a larger historical context is offset by the sparser frame rate. In Table II we evaluate our approach using the experiment settings presented in [13], which feature an extended history of 40 s and also allow to predict 20 trajectory candidates, which naturally leads to lower overall errors. Evaluation is done on the full TrajAir dataset (*111Days* split) using the standard prediction horizon of 120 s. Again, our approach achieves favorable results compared to all methods. The results highlight that ASCENT significantly improves the state-of-the-art on TrajAir across all experiment settings used by related work. Figure 4 shows qualitative results for scenarios taken from the TrajAir dataset *111Days* test set. We show the predictions of our ASCENT model for all aircraft present in a scene, as well as the historical data

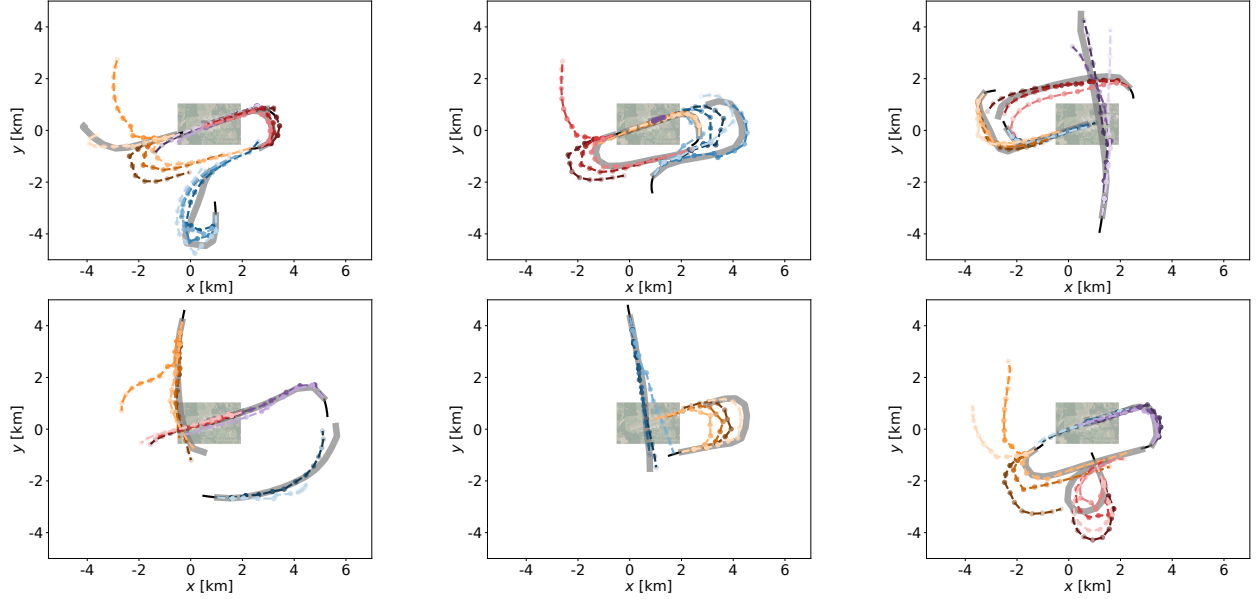


Fig. 4. Qualitative results on scenarios from the TrajAir [8] 111Days validation set. Visualizations show 11 s of historical aircraft movement (black), 120 s of ground truth future (gray) and predictions from our model (colored). For clearer visualization we exclude predictions with scores below 0.1.

TABLE II

PREDICTION RESULTS ON THE FULL TRAJAIR TEST SET (111Days). UPPER GROUP: FOLLOWING THE SETUP OF [13], ALL MODELS USE $h=40$ s INPUT AND PREDICT $K=20$ TRAJECTORIES. LOWER GROUP: ADDITIONAL RESULTS FOR OUR MODEL USING THE STANDARD SETUP FROM THE DATASET PAPER [8]. ERRORS REPORTED IN KM.

Input: 40 s Method	111Days	
	minADE ₂₀	minFDE ₂₀
Constant Velocity	1.85	4.16
TransformerTF [45]	1.67	3.94
Nearest Neighbor	1.97	2.70
STG-CNN [46]	1.37	2.35
S-PEC [51]	1.04	2.18
DAG-Net [52]	0.77	1.61
TrajAirNet [8]	0.79	1.58
SocialPatterNN-ATT [10]	0.67	1.40
PECNet [53]	0.67	1.14
MID [32]	0.55	0.87
Expert-Traj [47]	0.55	0.72
GoDFlight [13]	0.29	0.39
ASCENT (Ours)	0.19	0.26

Input: 11 s Method	111Days	
	minADE ₅	minFDE ₅
ASCENT (Ours)	0.29	0.49

and ground truth future. The visualizations highlight that our model yields accurate trajectory outputs while also capturing different motion modes. Table III provides results on the TartanAviation dataset, where we evaluate a constant velocity baseline model and our approach. We present results for training and testing on both splits, results marked with S1 mean testing is done on split S1 and vice-versa. Our model achieves strong results on the KBTP airport, as well as, on the data from the towered KAGC airport.

A. Ablation Study

For all ablation studies, we follow the experimental setup of TrajAirNet [8]: model input of 11 s and prediction of $k=5$ hypothesis for the following 120 s. Table IV presents an ablation study evaluating three key components of our

TABLE III

TRAJECTORY PREDICTION RESULTS ON THE TARTANAVIATION DATASET [9]. THE DATASET CONTAINS ADDITIONAL DATA FROM THE KBTP AIRPORT (SAME LOCATION AS TRAJAIR DATASET) AS WELL AS DATA FROM THE TOWERED KAGC AIRPORT. ERRORS REPORTED IN KM.

Input: 11 s Method	KBTP S1		KBTP S2	
	minADE ₅	minFDE ₅	minADE ₅	minFDE ₅
Constant Velocity	1.83	3.91	1.77	3.80
ASCENT (Ours)	0.36	0.60	0.32	0.56

Input: 11 s Method	KAGC S1		KAGC S2	
	minADE ₅	minFDE ₅	minADE ₅	minFDE ₅
Constant Velocity	1.77	3.78	1.86	3.96
ASCENT (Ours)	0.47	0.72	0.50	0.75

approach: (1) decoder module, (2) input/output coordinate-frame modeling, and (3) prediction format (direct positions vs. parameterized trajectories). First, we compare a baseline CVAE-based decoder [28], similar to the one used in TrajAirNet [8], against our proposed mode query-based decoder. Our decoder consistently outperforms the CVAE baseline, demonstrating that a mode query-based approach is better suited for the aircraft problem setting. Second, we investigate different input and output coordinate representations. We evaluate models using global coordinates, and locally normalized coordinates with or without global positional embeddings. Using only local motion features yields the weakest performance, since the model lacks global context. Directly using global coordinates performs well, but the best results are obtained with local normalization combined with positional embeddings; this way the model has relative motion structure and global reference. Finally, we compare two prediction formats: direct decoding of 3D positions versus parameterized trajectory decoding. For long-horizon forecasting, evaluated by the minimum final displacement error (minFDE), the parameter-based approach achieves more accurate results, producing more stable trajectories.

Table V presents cross-dataset results, where we train

TABLE IV

ABLATION STUDY ON USING DIFFERENT DECODER MODULES (A CVAE-BASED BASELINE AND OUR MODE QUERY-BASED MODEL) AND OUTPUT PREDICTION STRATEGIES (DIRECTLY PREDICTING XYZ-POSITIONS VS. PREDICTING FLIGHT PARAMETERS). FURTHERMORE, WE COMPARE DIFFERENT COORDINATE MODELINGS: GLOBAL COORDINATES VS. LOCAL COORDINATES WITH DIFFERENT TYPES OF NORMALIZATION. ERRORS GIVEN IN KM.

Decoder Architecture	Coordinate System	Normalization		Positional Embeddings	Flight Parameter Prediction	7Days-Avg	
		Positional	Angular			minADE ₅	minFDE ₅
CVAE Baseline	Local	✓	✓	✗	✗	0.68	1.33
Mode Queries	Local	✓	✓	✗	✗	0.52	0.98
CVAE Baseline	Global	-	-	-	-	0.48	0.82
CVAE Baseline	Local	✓	✓	✓	✗	0.44	0.80
Mode Queries	Global	-	-	-	-	0.38	0.60
Mode Queries	Local	✓	✗	✓	✗	0.36	0.60
Mode Queries	Local	✓	✓	✓	✗	0.35	0.60
Mode Queries	Local	✓	✗	✓	✓	0.37	0.58
Mode Queries	Local	✓	✓	✓	✓	0.35	0.58

TABLE V

CROSS-DATASET EVALUATION USING THE TRAJAIR AND TARTANAVIATION (TA) DATASETS. *TRAJECTORIES EXCEEDING THE TRAJAIR DATASET RECORDING RADIUS ARE FILTERED OUT.

Train Dataset	Test Dataset	minADE ₅	minFDE ₅
TrajAir 111days	TA KBTP S1	0.69	1.25
TrajAir 111days	TA KBTP S2	0.64	1.19
TrajAir 111days	TA KBTP S1*	0.33	0.58
TrajAir 111days	TA KBTP S2*	0.36	0.60
TA KBTP S1	TrajAir 111days	0.32	0.57
TA KBTP S2	TrajAir 111days	0.33	0.57

TABLE VI

INFERENCE LATENCY ANALYSIS ON A SINGLE NVIDIA V100 GPU ACROSS DIFFERENT NUMBERS OF AIRCRAFT (B).

Method	Latency in ms				
	$B = 1$	$B = 4$	$B = 8$	$B = 16$	$B = 32$
TrajAirNet [8]	57	118	191	350	673
ASCENT (Ours)	11	13	14	18	17

on either the TrajAir [8] or TartanAviation [9] dataset and evaluate on the other. Notably, TartanAviation spans a larger recording radius, whereas TrajAir restricts data to within 5 km of the runway. Training on TartanAviation and evaluating on TrajAir (*111Days* split) yields strong results. In contrast, training on TrajAir and evaluating on TartanAviation leads to issues with distant trajectories, as these are not represented in the TrajAir training data. This is confirmed by the fact that cross-dataset performance aligns closely when evaluation is restricted to trajectory ranges present in TrajAir.

B. Latency Analysis

Table VI compares the latency of ASCENT against TrajAirNet [8], to our knowledge the only related work with a public implementation. Even without optimization techniques, ASCENT achieves significantly lower latency across all batch sizes (*i.e.*, number of aircraft). This performance stems from two primary architectural advantages: the direct prediction of k hypotheses, which eliminates iterative decoder sampling, and a streamlined transformer- and MLP-based design. These results demonstrate that our approach is well-suited for edge deployment, providing a viable path toward real-time conflict detection to enhance safety in GA.

VI. CONCLUSIONS

We present a transformer-based trajectory prediction model that achieves state-of-the-art results on both the

TrajAir [8] and TartanAviation [8] datasets. By decomposing the encoding into local motion and global context, and introducing a mode query-based decoding of flight parameters, our model achieves highly accurate multi-modal predictions. A limitation of our approach is the implicit learning of global relationships, such as runway locations, which affect generalization to other airports without finetuning. This is primarily due to the limited availability of data from diverse non-towered airspaces, which constrains broader training and evaluation. Currently, we do not use additional information, such as weather conditions. However, future work can easily extend our model design to encompass such features.

REFERENCES

- [1] Federal Aviation Administration (FAA), “General Aviation Safety,” 2025.
- [2] M. E. Johnson and Y. Gu, “Estimating airport operations at General Aviation airports using the FAA NPIAS airport categories,” *International Journal of Aviation, Aeronautics, and Aerospace*, vol. 4, no. 1, 2017.
- [3] Federal Aviation Administration (FAA), “Airplane Flying Handbook (FAA-H-8083-3C) Chapter 8: Airport Traffic Patterns,” 2022.
- [4] National Transportation Safety Board (NTSB), “US Civil Aviation Accident Dashboard: 2008–2023,” 2023.
- [5] Z. Chen, D. Guo, and Y. Lin, “A Deep Gaussian Process-Based Flight Trajectory Prediction Approach and Its Application on Conflict Detection,” *Algorithms*, vol. 13, no. 11, 2020.
- [6] Y. Lin, J.-w. Zhang, and H. Liu, “Deep Learning Based Short-Term Air Traffic Flow Prediction Considering Temporal-Spatial Correlation,” *Aerospace Science and Technology*, vol. 93, 2019.
- [7] Z. Wang, M. Liang, and D. Delahaye, “A Hybrid Machine Learning Model for Short-Term Estimated Time of Arrival Prediction in Terminal Manoeuvring Area,” *Transportation Research Part C: Emerging Technologies*, vol. 95, 2018.
- [8] J. Patrikar, B. Moon, J. Oh, and S. Scherer, “Predicting Like A Pilot: Dataset and Method to Predict Socially-Aware Aircraft Trajectories in Non-Towered Terminal Airspace,” in *Proc. of the International Conference on Robotics and Automation (ICRA)*, 2022.
- [9] J. Patrikar, J. Dantas, B. Moon, M. Hamidi, S. Ghosh, N. Keetha, I. Higgins, A. Chandak, T. Yoneyama, and S. Scherer, “Image, Speech, and ADS-B Trajectory Datasets for Terminal Airspace Operations,” *Scientific Data*, vol. 12, no. 1, 2025.
- [10] I. Navarro and J. Oh, “Social-PatteRNN: Socially-Aware Trajectory Prediction Guided by Motion Patterns,” in *Proc. of the International Conference on Intelligent Robots and Systems (IROS)*, 2022.
- [11] Y. Yin, S. Zhang, Y. Zhang, Y. Zhang, and S. Xiang, “Context-Aware Aircraft Trajectory Prediction With Diffusion Models,” in *Proc. of the International Conference on Intelligent Transportation Systems (ITSC)*, 2023.
- [12] Y. Yin, S. Zhang, Y. Zhang, Y. Zhang, and S. Xiang, “Aircraft Trajectory Prediction in Terminal Airspace With Intentions Derived From Local History,” *Neurocomputing*, 2025.

- [13] S. Yang, L. Liu, B. Chen, S. Cheng, Z. Shi, and Z. Zou, "GooDFlight: Goal-Oriented Diffusion Model for Flight Trajectory Prediction," *IEEE Transactions on Aerospace and Electronic Systems*, 2025.
- [14] S. Shi, L. Jiang, D. Dai, and B. Schiele, "Motion Transformer with Global Intention Localization and Local Movement Refinement," in *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, 2022.
- [15] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, "Wayformer: Motion Forecasting via Simple & Efficient Attention Networks," in *Proc. of the International Conference on Robotics and Automation (ICRA)*, 2023.
- [16] X. Wang, T. Su, F. Da, and X. Yang, "ProphNet: Efficient Agent-Centric Motion Forecasting with Anchor-Informed Proposals," in *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [17] Z. Zhou, J. Wang, Y.-H. Li, and Y.-K. Huang, "Query-Centric Trajectory Prediction," in *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [18] Z. Lan, Y. Jiang, Y. Mu, C. Chen, S. E. Li, H. Zhao, and K. Li, "SEPT: Towards Efficient Scene Representation Learning for Motion Prediction," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2023.
- [19] J. Cheng, X. Mei, and M. Liu, "Forecast-MAE: Self-supervised Pre-training for Motion Forecasting with Masked Autoencoders," in *Proc. of the Conference on Computer Vision (ICCV)*, 2023.
- [20] S. Shi, L. Jiang, D. Dai, and B. Schiele, "MTR++: Multi-Agent Motion Prediction with Symmetric Scene Modeling and Guided Intention Querying," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 46, no. 5, 2024.
- [21] B. Zhang, N. Song, and L. Zhang, "DeMo: Decoupling Motion Forecasting into Directional Intentions and Dynamic States," in *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, 2024.
- [22] N. Song, B. Zhang, X. Zhu, and L. Zhang, "Motion Forecasting in Continuous Driving," in *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, 2024.
- [23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, 2017.
- [24] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "VectorNet: Encoding HD Maps and Agent Dynamics From Vectorized Representation," in *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [25] Y. Zhou, H. Shao, L. Wang, S. L. Waslander, H. Li, and Y. Liu, "SmartRefine: A Scenario-Adaptive Refinement Framework for Efficient Motion Prediction," in *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [26] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal Convolutional Networks for Action Segmentation and Detection," in *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [27] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph Attention Networks," *Proc. of the International Conference on Learning Representations (ICLR)*, 2018.
- [28] X. Yan, J. Yang, K. Sohn, and H. Lee, "Attribute2Image: Conditional Image Generation from Visual Attributes," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2016.
- [29] L. Verlet, "Computer Experiments on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules," *Physical Review*, vol. 159, no. 1, 1967.
- [30] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A Recurrent Latent Variable Model for Sequential Data," in *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, 2015.
- [31] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2020.
- [32] T. Gu, G. Chen, J. Li, C. Lin, Y. Rao, J. Zhou, and J. Lu, "Stochastic Trajectory Prediction via Motion Indeterminacy Diffusion," in *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [33] P. Xu, J.-B. Hayet, and I. Karamouzas, "SocialVAE: Human Trajectory Prediction using Timewise Latents," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2022.
- [34] D. Rempe, Z. Luo, X. Bin Peng, Y. Yuan, K. Kitani, K. Kreis, S. Fidler, and O. Litany, "Trace and Pace: Controllable Pedestrian Animation via Guided Trajectory Diffusion," in *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [35] G. Capellera, A. Rubio, L. Ferraz, and A. Agudo, "Unified Uncertainty-Aware Diffusion for Multi-Agent Trajectory Modeling," in *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.
- [36] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou, et al., "Large Scale Interactive Motion Forecasting for Autonomous Driving: The WAYMO OPEN MOTION DATASET," in *Proc. of the Conference on Computer Vision (ICCV)*, 2021.
- [37] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A Multimodal Dataset for Autonomous Driving," in *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [38] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "MultiPath: Multiple Probabilistic Anchor Trajectory Hypotheses for Behavior Prediction," in *Proc. of the Conference on Robot Learning (CoRL)*, 2020.
- [39] J. Mercat, T. Gilles, N. El Zoghby, G. Sandou, D. Beauvois, and G. P. Gil, "Multi-Head Attention for Multi-Modal Joint Vehicle Motion Forecasting," in *Proc. of the International Conference on Robotics and Automation (ICRA)*, 2020.
- [40] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, "Learning Lane Graph Representations for Motion Forecasting," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2020.
- [41] X. Jia, P. Wu, L. Chen, Y. Liu, H. Li, and J. Yan, "HDGT: Heterogeneous Driving Graph Transformer for Multi-Agent Trajectory Prediction via Scene Encoding," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2023.
- [42] A. Cui, S. Casas, K. Wong, S. Suo, and R. Urtasun, "GoRela: Go Relative for Viewpoint-Invariant Motion Forecasting," in *Proc. of the International Conference on Robotics and Automation (ICRA)*, 2023.
- [43] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, "Multimodal Motion Prediction with Stacked Transformers," *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [44] A. Prutsch, H. Bischof, and H. Possegger, "Efficient Motion Prediction: A Lightweight & Accurate Trajectory Prediction Model With Fast Training and Inference Speed," in *Proc. of the International Conference on Intelligent Robots and Systems (IROS)*, 2024.
- [45] F. Giuliari, I. Hasan, M. Cristani, and F. Galasso, "Transformer Networks for Trajectory Forecasting," in *Proc. of the International Conference on Pattern Recognition (ICPR)*, 2021.
- [46] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-STGCNN: A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction," in *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [47] H. Zhao and R. P. Wildes, "Where Are You Heading? Dynamic Trajectory Prediction With Expert Goal Examples," in *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [48] Y. Zhong, Z. Ni, S. Chen, and U. Neumann, "Aware of the History: Trajectory Forecasting with the Local Behavior Data," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2022.
- [49] H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, J. Schneider, D. Bradley, and N. Djuric, "Deep Kinematic Models for Kinetically Feasible Vehicle Trajectory Predictions," in *Proc. of the International Conference on Robotics and Automation (ICRA)*, 2020.
- [50] P. J. Huber, "Robust Estimation of a Location Parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, 1964.
- [51] D. Zhao and J. Oh, "Noticing Motion Patterns: A Temporal CNN With a Novel Convolution Operator for Human Trajectory Prediction," *IEEE Robotics and Automation Letters (RAL)*, 2020.
- [52] A. Monti, A. Bertugli, S. Calderara, and R. Cucchiara, "DAG-Net: Double Attentive Graph Neural Network for Trajectory Forecasting," in *Proc. of the International Conference on Pattern Recognition (ICPR)*, 2021.
- [53] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, "It Is Not the Journey But the Destination: Endpoint Conditioned Trajectory Prediction," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2020.