Context-aware Aircraft Trajectory Prediction with Diffusion Models

Yifang Yin, Sheng Zhang, Yicheng Zhang, Yi Zhang, and Shili Xiang

***Abstract*— Aircraft trajectory prediction aims to estimate the future movements of the aircraft, which is a crucial step for air traffic management such as capacity estimation and conflict detection. In this paper, we present a context-aware trajectory prediction method, which generates the future movements based on both the aircraft's past status and the contextual information such as the pilot and controller intent and the environmental conditions. The proposed framework consists of 1) a *Trajectory Encoder* that captures the history behaviors and the social interactions of the aircraft, 2) a *Context Encoder* that extracts latent features from contextual information, and 3) a *Transformer-based Decoder* that generates future trajectories based on a diffusion model. Specifically, we model the trajectory prediction as the reverse diffusion process where we first gradually add noise to the ground-truth trajectory and then train a neural network to learn the reverse of this diffusion process conditioned on the output of the trajectory encoder and the context encoder. We conduct experiments on real-world aircraft trajectories collected at Singapore Changi Airport in December 2019, which correspond to one-week ADS-B data before the start of the COVID-19 pandemic. The experimental results show that our proposed approach outperforms existing methods by a significant margin.**

I. INTRODUCTION

In recent years, the demand for air transport has been growing rapidly, which brings new challenges to the existing air traffic management systems [26]. The airports' capacity is under strain and the contradiction between aviation demand and airspace capacity has become increasingly prominent. To resolve this crisis, operators are focused on two directions simultaneously: growing capacity by physical expansion of airports and maximising existing capacity by intelligent planning. One critical component for the intelligent capacity planning is the accurate prediction of the future trajectory of the aircraft, which can bring substantial benefits for downstream tasks such as flight flow management, arrival and departure sequencing, conflict detection [24], *etc*.

Trajectory prediction is a critical yet challenging task, where various factors need to be considered simultaneously to model the uncertainty of the future movements. Fortunately, large-scale aircraft trajectories can now be easily collected from a few open databases to conduct trajectory prediction research. For example, the historical ADS-B data are available from platforms such as Flightradar24, FlightAware, and the OpenSky network. Public benchmark datasets for aircraft trajectory prediction, *e.g.*, TrajAir dataset [19], have also been presented recently. This motivates the development of data-driven aircraft trajectory prediction methods based on machine learning models. In such methods, the temporal dependencies of the observed trajectory (*i.e.* historical trajectory) are usually captured by the Hidden Markov Model (HMM) [2], Long Short-Term Memory (LSTM) [10], [29], or hybrid networks [12], [21]. While early learning-based methods mostly focus on trajectory prediction along a single route, more recent efforts have been made on the modeling of complex scenes with dynamic contexts [15], [18]. For instance, TrajAirNet [19] jointly predicts the future trajectories of multiple agents in the terminal airspace around airports. It not only captures the temporal dependencies of agents' observed trajectories, but also models the agent-agent social interactions and environmental contexts such as weather velocity and direction. However, compared to pedestrian and vehicle behavior analysis, research on aircraft trajectory prediction is still in its early stage.

In this paper, we present the first work that adopts the diffusion model for aircraft trajectory prediction. The original motion indeterminacy diffusion model was proposed for pedestrians [6], which is adapted to the aviation domain in the following three aspects. First, the 2D trajectories are extended to 3D to incorporate altitudes in both the input and output. Second, unlike previous work that normalized trajectories based on their respective current location, we use the absolute locations that correlate better with aircraft's future trajectories in the airspace. Third, we extract and investigate two types of contextual information that have not been utilized in previous work. Specifically, we extract two one-hot encodings, *i.e.*, one is based on the scheduled landing runway and direction, and the other is based on the estimated arrival time. Such information partially reveals the pilot and controller intent and thus helps improve the model's prediction accuracy. Latent features are extracted from the historical trajectories and the contextual information based on separate encoders, which are next concatenated and leveraged as the condition for generating future trajectories based on a diffusion model. The diffusion model is a generative model. It consists of 1) a forward process (or diffusion process) where the ground-truth trajectories are progressively noised with a pre-defined variance scheduler, and 2) a backward process (or reverse diffusion process) where the neural network learns to transform the noise back into a sample that follows the data distribution. Therefore, once the model is trained, we can generate the future trajectory from a random Gaussian noise recursively in K steps, where

Yifang Yin, Sheng Zhang, Yicheng Zhang, Yi Zhang, and Shili Xiang are with the Institute for Infocomm Research (I^{2}R), Agency for Science, Technology and Research (A*STAR), 1 Fusionopolis Way, #21-01 Connexis, Singapore 138632, Republic of Singapore. Email: {yin_yifang, zhang_sheng, zhang_yicheng, zhang_yi, sxiang}@i2r.a-star.edu.sg

This work is supported by the National Research Foundation, Singapore, and the Civil Aviation Authority of Singapore (CAAS), under the Aviation Transformation Programme (Grant No. ATP_IOP for ATM_I2R_2).

K controls the trade-off between diversity and determinacy of the generated trajectories.

The rest of this paper is organized as follows. Section II reports important related work. Section III introduces the technical details of our proposed context-aware aircraft trajectory prediction framework. Section IV illustrates qualitative and quantitative experimental results and verifies the effectiveness of our proposed method by comparing it to the state-of-the-arts. Finally, Section V draws the conclusion of this work.

II. RELATED WORK

A. Trajectory Prediction for Pedestrians and Vehicles

Trajectory prediction in the 2D space has been extensively studied for pedestrians and vehicles [4], [16], [25]. For instance, Social-LSTM [1] adopts LSTM models to encode historical trajectories and the interaction between neighbors. mmTransformer [11] leverages stacked transformers that consist of a motion extractor, a map aggregator, and a social constructor to integrate different contextual information hierarchically. Trajectron++ [20] presents a graph-based model to forecast the trajectories of diverse agents by incorporating agent dynamics and heterogeneous data. Zhao and Wildes [28] propose to first query potential goal positions from a repository of expert trajectories, and then generate the future trajectories based on the retrieved goals. While most existing work employs generative adversarial nets (GANs) [5] or conditional variational auto-encoder (CVAE) [23] to encode the multi-modal distribution of future trajectories, a recent work innovatively models trajectory prediction by motion indeterminacy diffusion (MID), which gradually reduces the indeterminacy from all reachable areas to the determinate prediction with a parameterized Markov chain [6]. However, all these methods focus on trajectory prediction in the 2D space, which performs unsatisfactorily for aircraft without adaptation.

B. Trajectory Prediction for Aircraft

Early work on aircraft trajectory prediction focuses on a single route and models the trajectories based on either physical dynamics [7], [13] or machine learning techniques including Hidden Markov Model (HMM) [2], Long Short-Term Memory (LSTM) [10], [29], CNN-LSTM [12], [17] or CNN-GRU (Gated Recurrent Units) [21] hybrid networks. Recent work considers more complex scenes and predicts dense future trajectories every few seconds [9], [15], [18], [19]. For instance, TrajAirNet [19] presents an end-to-end socially-aware trajectory prediction framework that consists of three major components: 1) a Temporal Convolutional Networks (TCNs) to encode the observed trajectories of each agent, 2) a Graph Attention Network (GATs) to model the agent-agent social interactions, and a Conditional Variational Autoencoder (CVAE) to encode the distribution of future trajectories. Social-PatteRNN [15] adopts a recurrent version of CVAE, *i.e.*, the Variational Recurrent Neural Network (VRNN) [3]. Inspired by DAG-Net [14], Social-PatteRNN also learns short-term motion patterns and social interactions from the observed trajectories, which are defined as the context features to condition the VRNN.

III. AIRCRAFT TRAJECTORY PREDICTION

A. Problem Formulation

Aircraft trajectory prediction aims at forecasting the future trajectories for aircraft based on their prior movements and other contextual information such as landing direction. Let $\mathbf{x}^i = \{s_t^i \in \mathbb{R}^3 | t = -H, -H+1, \dots, 0\}$ denote the prior movements for the i-th aircraft in a scene at time $t = 0$, where s_t^i is the 3D location at time t and H is the length of the historical trajectory. Let $\mathbf{c}^i = \{c^i_{time}, c^i_{dirc}\}$ denote the contextual information *w.r.t.* the estimated landing time and landing direction. Given $\{\mathbf{x}^i\}$ and $\{\mathbf{c}^i\}$ of all the aircraft in a scene, our goal is to predict their future trajectories represented by $\mathbf{y}^i = \{s_t^i \in \mathbb{R}^3 | t = 1, 2, \dots, T\}$ where T is the length of the future trajectory to be predicted.

B. Motion Indeterminacy Diffusion

The motion indeterminacy diffusion model was originally proposed by Gu *et al.* for pedestrian trajectory prediction [6]. The idea is to define a diffusion process for every aircraft as $(\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_K)$[1], where $\mathbf{y}_0$ is the ground-truth trajectory under the data distribution, $\mathbf{y}_K$ is the ambiguous region under the noise distribution, and K is the maximum number of diffusion steps. Specifically, this diffusion process gradually adds Gaussian noise to the ground-truth trajectory $\mathbf{y}_0$ until it corrupted into an ambiguous region $\mathbf{y}_K$ by,

$$\begin{aligned} q(\mathbf{y}_{1:K}|\mathbf{y}_0) &:= \prod_{k=1}^{K} q(\mathbf{y}_k|\mathbf{y}_{k-1}) \\ q(\mathbf{y}_k|\mathbf{y}_{k-1}) &:= \mathcal{N}(\mathbf{y}_k; \sqrt{1-\beta_k}\mathbf{y}_{k-1}, \beta_k \mathbf{I}) \end{aligned} \tag{1}$$

where $\beta_1, \beta_2, \dots, \beta_K$ are fixed variance schedulers that control the scale of the injected noise. With reparameterization, we can compute $\mathbf{y}_k$ at any diffusion step k based on $\mathbf{y}_0$ in a closed form as,

$$q(\mathbf{y}_k|\mathbf{y}_0) := \mathcal{N}(\mathbf{y}_k; \sqrt{\overline{\alpha}_k}\mathbf{y}_0, (1-\overline{\alpha}_k)\mathbf{I}) \tag{2}$$

where $\alpha_k = 1 - \beta_k$ and $\overline{\alpha}_k = \prod_{s=1}^{k} \alpha_s$. Therefore, after sufficient diffusion steps of recursively adding noise, we obtain a $\mathbf{y}_K$ that approximately follows the Gaussian noise distribution $\mathcal{N}(0, \mathbf{I})$.

Next, the trajectory prediction problem is defined as the reverse diffusion process from noise distribution, conditioned on both the historical trajectory $\mathbf{x}$ and the contextual information $\mathbf{c}$. We extract the condition representation as $\mathbf{f} = \mathcal{F}_x(\mathbf{x}) \| \mathcal{F}_c(\mathbf{c})$, where $\mathcal{F}_x$ is a temporal-social trajectory encoder, $\mathcal{F}_c$ is a context encoder, and $\|$ is the concatenation operator. Thereafter, the reverse diffusion process is formulated based on Markov chain as,

$$\begin{aligned} p_\theta(\mathbf{y}_{0:K}|\mathbf{f}) &:= p(\mathbf{y}_K) \prod_{k=1}^{K} p_\theta(\mathbf{y}_{k-1}|\mathbf{y}_k, \mathbf{f}) \\ p_\theta(\mathbf{y}_{k-1}|\mathbf{y}_k, \mathbf{f}) &:= \mathcal{N}(\mathbf{y}_{k-1}; \boldsymbol{\mu}_\theta(\mathbf{y}_k, k, \mathbf{f}), \boldsymbol{\Sigma}_\theta(\mathbf{y}_k, k)) \end{aligned} \tag{3}$$

[1]Here we use $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{c}$ without the superscript i for simplicity purposes.

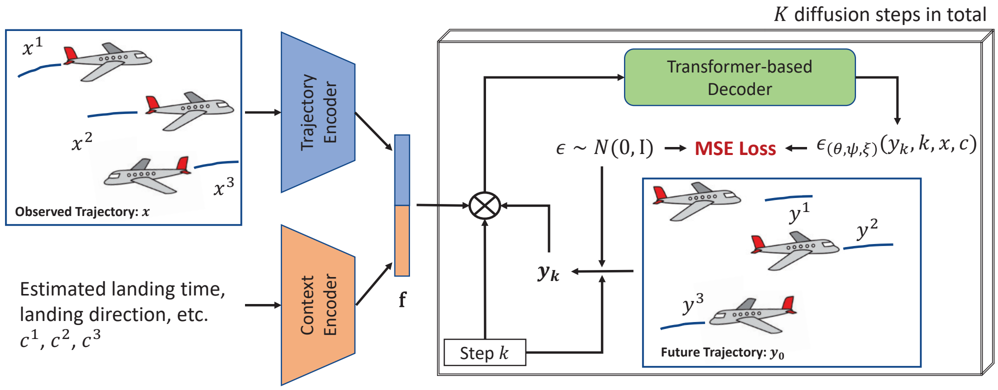

Fig. 1. Overview of our proposed context-aware aircraft trajectory prediction with diffusion models.

where $p(\mathbf{y}_K)$ is the initial noise Gaussian distribution. θ denotes the parameter of the diffusion model, which estimates the mean $\boldsymbol{\mu}_\theta$ and variance $\boldsymbol{\Sigma}_\theta$ of the Gaussian transition based on $\mathbf{y}_k$, k, and $\mathbf{f}$. Following previous work [8], [6], we set $\boldsymbol{\Sigma}_\theta(\mathbf{y}_k, k) = \beta_k \mathbf{I}$, and train the diffusion model, the trajectory encoder, and the context encoder to learn $\boldsymbol{\mu}_\theta$ using the trajectory data.

Training Objective: The overall network can be trained by minimizing the negative variational lower bound of the log-likelihood $\mathbb{E}[\log p_\theta(\mathbf{y}_0)]$ [22], [6]. The loss function is formally given as,

$$\begin{aligned} L =& \mathbb{E}_q[\sum_{k=2}^{K} D_{KL}(q(\mathbf{y}_{k-1}|\mathbf{y}_k, \mathbf{y}_0) \parallel p_\theta(\mathbf{y}_{k-1}|\mathbf{y}_k, \mathbf{f})) \\ &- \log p_\theta(\mathbf{y}_0|\mathbf{y}_1, \mathbf{f})] \end{aligned} \tag{4}$$

where the posterior $q(\mathbf{y}_{k-1}|\mathbf{y}_k, \mathbf{y}_0)$ can be written as $q(\mathbf{y}_{k-1}|\mathbf{y}_k, \mathbf{y}_0) := \mathcal{N}(\mathbf{y}_{k-1}; \tilde{\boldsymbol{\mu}}_k(\mathbf{y}_k, \mathbf{y}_0), \tilde{\beta}_k \mathbf{I})$. Recall that the reverse diffusion process $p_\theta(\mathbf{y}_{k-1}|\mathbf{y}_k, \mathbf{f})$ is also Gaussian (Eq. 3), we thus can calculate the first term D_{KL} by the difference between the means of $\tilde{\boldsymbol{\mu}}_k$ and $\boldsymbol{\mu}_\theta$ as,

$$D_{KL} = \mathbb{E}_q[\lambda \parallel \tilde{\boldsymbol{\mu}}_k(\mathbf{y}_k, \mathbf{y}_0) - \boldsymbol{\mu}_\theta(\mathbf{y}_k, k, \mathbf{f})\|^2] + C \tag{5}$$

where λ and C are coefficients with no effect on the gradient direction. Note that the second term $-\log p_\theta(\mathbf{y}_0|\mathbf{y}_1, \mathbf{f})$ can also be formulated as shown in Eq. 5 when $k = 1$. Following previous work [8], we apply the parameterization method to reparameterize,

$$\boldsymbol{\mu}_\theta(\mathbf{y}_k, k, \mathbf{f}) = \frac{1}{\sqrt{\alpha_k}}(\mathbf{y}_k - \frac{\beta_k}{\sqrt{1-\bar{\alpha}_k}} \epsilon_\theta(\mathbf{y}_k, k, \mathbf{f})) \tag{6}$$

and obtain a simplified loss function as,

$$L = \mathbb{E}_{\epsilon, \mathbf{y}_0, k} \parallel \epsilon - \epsilon_{(\theta,\psi,\xi)}(\mathbf{y}_k, k, \mathbf{x}, \mathbf{c}) \| \tag{7}$$

where θ, ψ, and ξ denote the parameters of the diffusion model, the trajectory encoder, and the context encoder, respectively. $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, $\mathbf{y}_k = \sqrt{\bar{\alpha}_k}\mathbf{y}_0 + \sqrt{1-\bar{\alpha}_k}\epsilon$, and the optimization is performed at each step $k \in [1, K]$.

Inference: Once the model is trained, we can generate the future trajectory $\mathbf{y}_0$ based on a random $\mathbf{y}_K$ drawn from $\mathcal{N}(0, \mathbf{I})$ recursively as,

$$\mathbf{y}_{k-1} = \frac{1}{\sqrt{\alpha}}(\mathbf{y}_k - \frac{\beta_k}{\sqrt{1-\bar{\alpha}_k}} \epsilon_{(\theta,\psi,\xi)}(\mathbf{y}_k, k, \mathbf{x}, \mathbf{c})) + \sqrt{\beta_k}\mathbf{z} \tag{8}$$

where $\mathbf{z}$ is a random variable in standard Gaussian distribution and $\epsilon_{(\theta,\psi,\xi)}$ is the trained network.

C. Model Architecture

As shown in Fig. 1, our proposed framework consists of three major components, namely the trajectory encoder, the context encoder, and the Transformer-based decoder for the diffusion model. Next, we introduce the technical details of each component.

1) Trajectory encoder: The historical trajectories of aircraft are given as $\mathbf{x}^i = \{s_t^i \in \mathbb{R}^3 | t = -H, -H+1, \ldots, 0\}$ where i denotes the i-th aircraft, s_t^i is the 3D location of the i-th aircraft at time t, and H is the length of the historical trajectory. We normalize the 3D locations given in format of latitude, longitude, and altitude with respect to the centre of the airport in kilometers. Most of the existing trajectory prediction methods for pedestrians and cars further convert this normalized absolute coordinates to relative coordinates by subtracting s_0^i from each $\mathbf{x}^i$. But we adopt the absolute coordinates as our input as it has been shown to be more effective for aircraft trajectory prediction [19]. For the trajectory encoder, we adopt the Trajectron++ encoder [20] for its superior representation ability. It represents a scene as a graph, where the nodes are agents and the edges represent their interactions. The agent history is encoded with LSTMs and the agent interactions are encoded by aggregating information from all the neighboring agents.

2) Context encoder: We extract two types of contextual information to improve the trajectory prediction for aircraft. First, airports usually have more than one runway, and pilots will be informed beforehand about the runway and direction they should follow for landing. We extract a one-hot encoding based on this information for each aircraft, denoted as c_{dirc}^i. Take an airport with two runways as an example, $c_{dirc}^i = [1, 0, 0, 0]$ or $c_{dirc}^i = [0, 1, 0, 0]$ represents that the i-th aircraft will be landing on the first runway from either of the two directions. Similarly, $c^i_{dirc} = [0, 0, 1, 0]$ or $c^i_{dirc} = [0, 0, 0, 1]$ represents that the i-th aircraft will be landing on the second runway. Second, we extract the time it takes from now (*i.e.*, $t = 0$) till landing, denoted as t^i_{ldg}, from our trajectory dataset. In real-world scenarios, t^i_{ldg} can be estimated based on the difference between the current time and the estimated landing time. However, the actual landing time can be a few minutes early or late compared to the estimated landing time. To simulate this situation, we add a random Guassian noise to t^i_{ldg} and quantize t^i_{ldg} into a one-hot encoding, denoted as c^i_{time}, where the duration of each bin is set to 1 minute. Thereafter, we use two separate embedding layers as the context encoder to process c^i_{dirc} and c^i_{time} and concatenate their outputs. It converts two sparse one-hot encodings to a dense vector representation.

3) Transformer-based decoder: Let $\mathcal{F}_x$ and $\mathcal{F}_c$ denote the trajectory encoder and context encoder introduced in the above sections. We compute $\mathbf{f} = \mathcal{F}_x(\mathbf{x}) \parallel \mathcal{F}_c(\mathbf{c})$, where $\parallel$ represents the concatenation operator. To model the Gaussian transitions in the reverse diffusion process, we adopt the Transformer-based decoder of MID [6]. It takes the ground-truth trajectory $\mathbf{y}_0$, the noise variable $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, the condition representation $\mathbf{f}$, and a time embedding as the input. In each step k, we first compute $\mathbf{y}_k$ by adding noise to $\mathbf{y}_0$ as $\mathbf{y}_k = \sqrt{\bar{\alpha}_k}\mathbf{y}_0 + \sqrt{1 - \bar{\alpha}_k}\epsilon$ and concatenate the time embedding with the condition $\mathbf{f}$ as contextual features. Next, we encode both $\mathbf{y}_k$ and the contextual features with fully-connected layers to generate a fused feature. A positional embedding is also generated and added to the fused feature to model the positional relation of $\mathbf{y}_k$ at different trajectory timestamp t. Finally, we feed the fused feature to the Transformer-based decoder to estimate $\epsilon_{(\theta,\psi,\xi)}(\mathbf{y}_k, k, \mathbf{x}, \mathbf{c})$. The mean square error (MSE) is adopted as the loss function.

IV. EVALUATION

A. Experimental Setup

Datasets: We conduct experiments on real-world flight trajectories collected at Singapore Changi Airport from December 1st 2019 to December 7th 2019, *i.e.*, one-week ADS-B data before the start of the COVID-19 pandemic. We consider the arrival flights only and divide the dataset into 6-day data (December 1st to 6th) for training and 1-day data for testing (December 7th). We further divide the trajectories into four subsets based on the different directions they enter the Singapore TMA. We visualize our experimental trajectory dataset in Fig. 2 and report the statistics of each subset in Table I.

Evaluation metrics: We adopt two widely-used metrics [27] for evaluating the performance of our proposed method, namely the Average Displacement Error (ADE) and the Final Displacement Error (FDE). ADE computes the average error between all the ground truth positions and the predicted positions in the trajectory, and FDE computes the displacement between the end points of ground truth and predicted trajectories. Results obtained based on Best-of-N strategy are reported where the trajectory prediction is performed N times, and the best ADE and FDE scores are recorded.

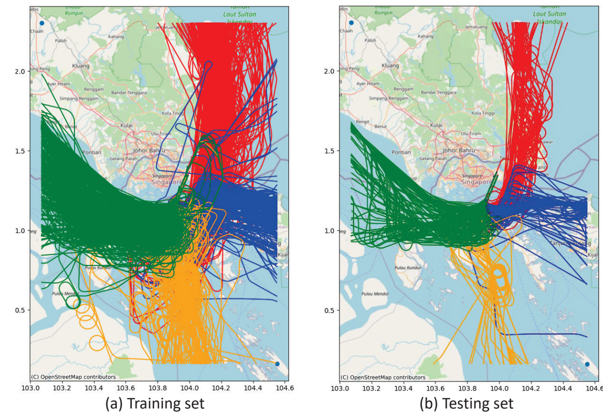

Fig. 2. Visualization of the experimental trajectory dataset within the Singapore Terminal Maneuvering Area (TMA).

TABLE I

STATISTICS OF THE EXPERIMENTAL TRAJECTORY DATASET.

subsets	North	East	South	West	Overall
No. of training traj.	1141	660	341	961	3103
No. of testing traj.	197	119	49	156	521

Implementation details: We sample the trajectories with 5 seconds interval, where the first 16 seconds (*i.e.*, 4 points including the current location) of a trajectory is used as historical trajectory to predict the next 2 minutes (*i.e.*, 24 points) future trajectory. As introduced in Section III-C, we use the normalized absolute coordinates of the historical trajectory as the input to predict the future velocities, which are next integrated to obtain the final absolute future trajectory. In our proposed network, the trajectory encoder encodes the historical trajectories and their interactions into a 256-d vector. The context encoder encode the contextual information into a 8-d vector. The dimension of the Transformer-based decoder is set to 512, which is progressively downsampled by fully-connected layers from 512-d to 3-d at each timestamp to match the dimension of the ground-truth trajectory. For optimization, we adopt a learning rate of 0.001 and a batch size of 256, and train our model with the Adam optimizer. All experiments were conducted on a single 24 GB NVIDIA GeForce GPU. Following previous work [19], we set N=5 for the Best-of-N evaluation strategy.

B. Comparison with the state-of-the-art methods

To evaluate the effectiveness of our proposed approach, we compare our method to the following five baselines for both 2D and 3D trajectory prediction:

1) Constant Velocity: We assume that the velocity does not change and set the acceleration to zero to predict the future trajectory.
2) Nearest Neighbour: Based on the Euclidean distance between historical trajectories, we predict the future trajectory of a target agent by finding its closest neighbour in the training dataset.
3) Trajectron++ [20]: A graph-structured recurrent model that forecasts the trajectories while incorporating agent

TABLE II

QUANTITATIVE RESULTS ON 2D AIRCRAFT TRAJECTORY PREDICTION WITH BEST-OF-5 STRATEGY IN ADE AND FDE METRICS.

Method	Absolute Location	North		East		South		West		AVG	
		ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
Constant Velocity	-	1.687	4.030	1.748	4.237	1.648	3.974	1.660	3.946	1.686	4.047
Nearest Neighbor	✓	1.170	2.380	1.408	2.783	1.716	3.494	1.435	2.889	1.432	2.887
Trajectron++ [20]	✗	0.713	1.602	0.732	1.620	0.707	1.636	0.798	1.854	0.738	1.678
MID [6]	✗	0.602	1.267	0.665	1.427	0.643	1.379	0.688	1.488	0.650	1.390
Ours w/o context	✓	0.493	0.936	0.530	1.009	0.555	1.115	0.571	1.114	0.537	1.044
Ours	✓	0.468	0.880	0.497	0.904	0.521	1.010	0.533	1.020	0.508	0.962

TABLE III

QUANTITATIVE RESULTS ON 3D AIRCRAFT TRAJECTORY PREDICTION WITH BEST-OF-5 STRATEGY IN ADE AND FDE METRICS.

Method	Absolute Location	North		East		South		West		AVG	
		ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
Constant Velocity	-	1.697	4.048	1.760	4.259	1.661	3.998	1.672	3.968	1.698	4.068
Nearest Neighbor	✓	1.172	2.330	1.303	2.500	1.433	2.867	1.404	2.773	1.328	2.618
TrajAirNet [19]	✓	0.844	1.719	0.883	1.805	0.947	2.019	0.979	2.031	0.913	1.894
MID [6]	✗	0.612	1.281	0.694	1.466	0.668	1.447	0.699	1.490	0.668	1.421
Ours w/o context	✓	0.498	0.951	0.559	1.033	0.596	1.190	0.593	1.123	0.562	1.074
Ours	✓	0.468	0.883	0.530	0.972	0.539	1.051	0.575	1.104	0.528	1.003

dynamics and heterogeneous data.

4) MID [6]: A Transformer-based model that formulates the trajectory prediction task as a reverse process of motion indeterminacy.
5) TrajAirNet [19]: A graph-based socially-aware framework for aircraft trajectory prediction.

The results are reported in Tables II and III. For the 2D trajectory prediction, we ignore the altitude values in both historical and future trajectories. The Constant Velocity method serves as a naive baseline. It achieves an average ADE/FDE of 1.686/4.047 and 1.698/4.068 on the 2D and 3D aircraft trajectory prediction, respectively. The Nearest Neighbor method improves ADE/FDE to 1.432/2.887 and 1.328/2.618. It is worth noting that the Nearest Neighbor method performs better on 3D trajectory prediction. One reason might be that the nearest neighbor search result obtained by 3D locations is more reliable, leading to improved performance for trajectory forecasting. The Trajectron++ method and the MID method are originally proposed for pedestrian or vehicle trajectory prediction. Though both of them perform generally well, the use of relative coordinates and the ignoring of the contextual information limit their performance for aircraft trajectory prediction. The TrajAirNet is a recently proposed trajectory prediction method for aircraft. It uses absolute coordinates as the input and considers the fine-grained wind velocity and direction as the contextual information. Though promising results have been obtained on the TrajAir dataset, its generalization ability to forecast trajectories in a large scene (*e.g.*, Singapore TMA) seems to be limited. Comparatively, our proposed method outperforms all its competitors and reduces the average ADE and FDE significantly in both Table II and Table III, which verifies the effectiveness of our proposed approach.

C. Ablation study

We first evaluate the impact of the contextual information such as the estimated landing time and landing direction on the trajectory prediction of aircraft. We remove the context encoder from our proposed framework and compute the condition as $\mathbf{f} = \mathcal{F}_x(\mathbf{x})$. The prediction results obtained without the contextual information are also reported in Tables II and III. As it shows the ADE and FDE increased by 5.7% $\sim$ 8.5% w/o context, which verifies the importance of the contextual information in aircraft trajectory prediction.

TABLE IV

ADE/FDE COMPARISON WITH VARYING LENGTH OF THE HISTORICAL TRAJECTORY AS THE INPUT.

Len.	North	East	South	West	AVG
2 pts	0.52/0.98	0.54/1.01	0.55/1.09	0.61/1.14	0.56/1.06
3 pts	0.49/0.92	0.54/0.98	0.54/1.05	0.58/1.11	0.54/1.02
4 pts	0.47/0.88	0.53/0.97	0.54/1.05	0.58/1.10	0.53/1.00
5 pts	0.48/0.89	0.52/0.95	0.54/1.03	0.57/1.08	0.53/0.99
6 pts	0.47/0.87	0.52/0.95	0.53/1.03	0.56/1.07	0.52/0.98

Next, we study the impact of the historical trajectory length on the prediction results. In our previous experiments, we use 4 historical points with 5 seconds sampling interval, which correspond to a 16-second trajectory segment as the input. Here we evaluate our method using different number of historical points and report the results in Table IV. Generally speaking, improved trajectory prediction results can be obtained by using longer historical trajectory as the condition. However, as the future trajectory is more correlated with recent locations, the improvement becomes less significant when the length of the historical trajectory is larger than 4 points. We thus set the length of the historical trajectory to 4 throughout the other experiments.

D. Visualization

We further evaluate the effectiveness of our proposed framework by performing qualitative studies. Fig. 3 illustrates the multiple predicted trajectories (*i.e.*, the five dashed green lines in each image) on the Singapore TMA dataset. The observed historical trajectory and the ground-truth future

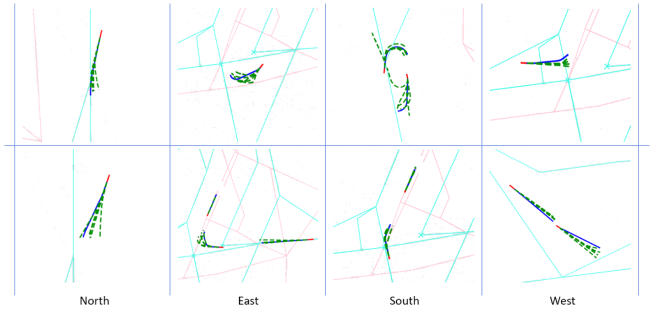

Fig. 3. Visualization of the predicted trajectories on our Singapore TMA dataset. Given the observed historical trajectories (red), we illustrate the ground-truth future paths (blue) and the predicted future trajectories by our method (dashed green).

trajectory are visualized in red and blue, respectively. The qualitative results show that our predicted trajectories fit the ground-truth path well. It indicates that our method is able to capture various behaviors of the aircraft including vectoring and holding when they enter the Singapore TMA from different directions.

V. CONCLUSIONS

We present the first context-aware diffusion-based framework for aircraft trajectory prediction, which consists of a trajectory encoder, a context encoder, and a diffusion-based decoder. The framework is designed based on our observations that the aircraft's behavior is closely related to its absolute location and domain-specific contexts such as the landing direction, estimated landing time, and/or weather information depending on its availability. We conduct extensive experiments on one-week ADS-B data collected at Singapore Changi Airport. The quantitative and qualitative results show that our method can capture various behaviors of the aircraft entering the Singapore TMA from different directions, which outperforms existing methods by a significant margin in terms of both ADE and FDE.

ACKNOWLEDGMENT

The authors would like to thank all colleagues from CAAS for providing valuable comments and suggestions on this work. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore and the Civil Aviation Authority of Singapore.

REFERENCES

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *CVPR*, pages 961–971, 2016.

[2] S. Ayhan and H. Samet. Aircraft trajectory prediction made easy with predictive analytics. In *ACM SIGKDD*, pages 21—-30, 2016.

[3] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio. A recurrent latent variable model for sequential data. *Advances in neural information processing systems*, 28, 2015.

[4] 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 *CVPR*, pages 11525–11533, 2020.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NeurIPS*, volume 27, pages 2672—-2680, 2014.

[6] T. Gu, G. Chen, J. Li, C. Lin, Y. Rao, J. Zhou, and J. Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *CVPR*, pages 17113–17122, 2022.

[7] A. M. Hernández, E. J. C. Magaña, and A. G. Berna. Data-driven aircraft trajectory predictions using ensemble meta-estimators. In *Digital Avionics Systems Conference*, pages 1–10, 2018.

[8] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020.

[9] P. Jia, H. Chen, L. Zhang, and D. Han. Attention-LSTM based prediction model for aircraft 4-D trajectory. *Scientific reports*, 12(1):1–11, 2022.

[10] Y. Liu and M. Hansen. Predicting aircraft trajectories: A deep generative convolutional recurrent neural networks approach. *arXiv preprint arXiv:1812.11670*, 2018.

[11] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou. Multimodal motion prediction with stacked transformers. In *CVPR*, pages 7577–7586, 2021.

[12] L. Ma and S. Tian. A hybrid CNN-LSTM model for aircraft 4D trajectory prediction. *IEEE access*, 8:134668–134680, 2020.

[13] S. Mondoloni and N. Rozen. Aircraft trajectory prediction and synchronization for air traffic management applications. *Progress in aerospace sciences*, 119:100640, 2020.

[14] A. Monti, A. Bertugli, S. Calderara, and R. Cucchiara. Dag-net: Double attentive graph neural network for trajectory forecasting. In *International Conference on Pattern Recognition*, pages 2551–2558, 2021.

[15] I. Navarro and J. Oh. Social-PatteRNN: Socially-aware trajectory prediction guided by motion patterns. In *International Conference on Intelligent Robots and Systems*, pages 9859–9864, 2022.

[16] B. Pang, T. Zhao, X. Xie, and Y. N. Wu. Trajectory prediction with latent belief energy-based model. In *CVPR*, pages 11814–11824, 2021.

[17] Y. Pang, H. Yao, J. Hu, and Y. Liu. A recurrent neural network approach for aircraft trajectory prediction with weather features from sherlock. In *AIAA Aviation 2019 Forum*, page 3413, 2019.

[18] J. Patrikar, J. Dantas, S. Ghosh, P. Kapoor, I. Higgins, J. J. Aloor, I. Navarro, J. Sun, B. Stoler, M. Hamidi, et al. Challenges in close-proximity safe and seamless operation of manned and unmanned aircraft in shared airspace. *arXiv preprint arXiv:2211.06932*, 2022.

[19] 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 *International Conference on Robotics and Automation*, pages 2525–2531, 2022.

[20] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *ECCV*, pages 683–700, 2020.

[21] N. Schimpf, E. J. Knoblock, Z. Wang, R. D. Apaza, and H. Li. A generalized approach to aircraft trajectory prediction via supervised deep learning. 2022.

[22] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265, 2015.

[23] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. *NeurIPS*, 28:3483—-3491, 2015.

[24] Y. Song, P. Cheng, and C. Mu. An improved trajectory prediction algorithm based on trajectory data mining for air traffic management. In *IEEE International Conference on Information and Automation*, pages 981–986, 2012.

[25] J. Sun, Y. Li, H.-S. Fang, and C. Lu. Three steps to multimodal trajectory prediction: Modality clustering, classification and synthesis. In *ICCV*, pages 13250–13259, 2021.

[26] W. Zeng, X. Chu, Z. Xu, Y. Liu, and Z. Quan. Aircraft 4D trajectory prediction in civil aviation: A review. *Aerospace*, 9(2):91, 2022.

[27] 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*, 6(2):628–634, 2020.

[28] H. Zhao and R. P. Wildes. Where are you heading? dynamic trajectory prediction with expert goal examples. In *ICCV*, pages 7629–7638, 2021.

[29] Z. Zhao, W. Zeng, Z. Quan, M. Chen, and Z. Yang. Aircraft trajectory prediction using deep long short-term memory networks. In *CICTP*, pages 124–135. 2019.