SafeAir: Safety-informed Prediction and Navigation under Distribution Shifts in General Aviation

1st Ricardo Gonzalo Cruz Castillo
School of Engineering and Sciences
Tec de Monterrey
Monterrey, Mexico
a01284147@tec.mx

2nd Ingrid Navarro
The Robotics Institute
Carnegie Mellon University
Pittsburgh, United States
ingridn@andrew.cmu.edu

3rd Jean Oh
The Robotics Institute
Carnegie Mellon University
Pittsburgh, United States
hyaejino@andrew.cmu.edu

***Abstract*— Data-driven trajectory prediction models are vital for downstream applications such as navigation in crowded environments. Recent developments have achieved promising results in characterizing agent-to-agent relationships and reasoning over multiple future outcomes. In the domain of aviation, where safety is critical, prediction models must be able to reason about complex scenarios that could lead to unsafe outcomes. However, existing works still fail to reason about safety due to design and training assumptions that limit their capabilities, such as short-sighted predictions and lack of situational awareness. As a result, prediction models will have poor performance in out-of-distribution cases since they lack critical context information and are therefore unreliable for deployment in downstream tasks such as navigation.**

With this in mind, we introduce `SafeAir`, a framework that explores and improves the out-of-distribution performance of navigation algorithms informed by data-driven trajectory prediction models. We address unexpected situations by creating a scenario hierarchy and using the most difficult ones for validation. `SafeAir` builds upon prior works that focus on safety-informed scenario characterization, to provide three main components: (1) A scenario scoring function based on safety measures. (2) a safety-informed motion prediction model that accounts for long-term outcomes and has a semantic understanding. (3) A navigation algorithm that uses the prediction model for decision-making and is tested on the scenario split obtained in (1). For testing, we create an air traffic control environment where collision checks are made throughout a goal-defined trajectory.

Our results show better prediction performance in out-of-distribution settings against state-of-the-art baselines.

I. Introduction

In crowded environments, a social robot must exhibit behavior that aligns with social norms to mitigate potential discomfort for humans coexisting within its proximity. This requires the robot to develop strong situational awareness, not only to adhere to social conventions but also to discern circumstances that may pose safety risks, enabling the adaptation of its behavior to avert unsafe outcomes. This is particularly crucial in safety-critical domains like aviation, which is the specific area of focus in this research.

Within the field of **aviation**, data-driven approaches for characterizing agent-to-agent interactions [1, 2, 3] have shown promising trajectory forecasting results. These methods have been extensively investigated in the domain of socially aware motion prediction due to their ability to capture social interactions purely through data. This approach

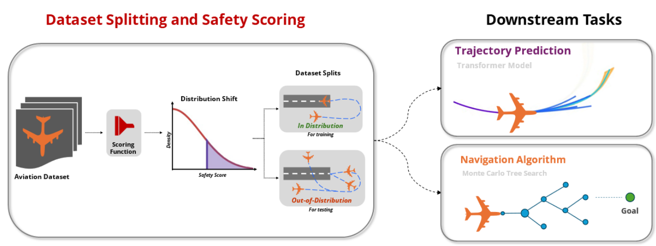

Fig. 1: Overview of `SafeAir`, a safety-informed scenario characterization framework for creating safety-informed distribution shifts and assessing the robustness of prediction and planning algorithms for **general aviation** under safety-relevant scenarios.

eliminates the need to develop custom reward functions or simulations, thereby simplifying the integration of social factors into motion prediction models. However, these methods alone will not perform well in downstream navigation tasks since they are myopic; they only address short-term interactions. Therefore, a need for robustification surfaces.

`SoRTS` [4] is an algorithm designed for social navigation that integrates a data-driven, socially-aware trajectory prediction model with a Monte Carlo Tree Search (MCTS) policy. Its structure addresses the issue of short-sighted predictions by directly incorporating social awareness into the search process.

However, data-driven trajectory prediction approaches are generally trained on datasets where the frequency of safety-critical scenarios is low, these scenarios can be regarded as out-of-distribution (OOD). The scarcity of critical scenarios in these 'benign' datasets affects the robustness of the prediction models, causing their performance to drop in unseen, critical scenarios, thus, making them unsafe for real-world deployment [5]. SoRTS is trained on the TrajAir[2] dataset, where these OOD scenes are not a common occurrence, thus it is not designed to perform well in safety-critical scenarios.

There has been previous work on assessing performance in safety-critical scenes, such as `SafeShift`[5]. Even though it is applied to the autonomous driving domain, it characterizes and scores scenarios based on safety-relevant metrics and creates distribution shifts based on score density across the dataset. Thus, a test bed is created for assessing robustness in OOD scenes.

By extending SafeShift to support TrajAir, we propose SafeAir, a framework that seeks to improve navigation performance in out-of-distribution scenarios. We introduce a context-aware transformer-based trajectory prediction model and a data-driven, long-horizon navigation approach. In this manner, offline and online model-enhancing techniques are applied ensuring robustness in unseen scenarios. Specifically, we focus on the aviation domain, where air traffic control datasets lack flight pattern information, and recording safety-critical scenarios would put human agents at risk.

II. Related Work

A. Distribution Shift Creation for Improved Robustness

In a real-world deployment, unseen, safety-critical scenarios will eventually present themselves. Real-world datasets usually lack safety-critical scenarios since they are not a common occurrence and it's not viable to create them. By assessing robustness in these settings, many state-of-the-art prediction models perform poorly because they lack context information and safety awareness [6, 7, 8]. Some recent works identify domains based on scene meta characteristics [9, 10] or clustering scenes into domains based on different features such as lane shape information, lane deflection angles, global bounds of scenario and trajectories, etc [11]. Thus the need for alleviating the impact of the distribution shift has been approached in multiple manners such as: by leveraging Frenet coordinates [11, 12], few-shot adaptation [9], or motion-based style transfer [13]. However, they differ from previous work [5] in that their distribution shifts are not based on safety-oriented scene characterization and scoring. In this framework, We further build upon this work by extending it into the aviation domain. We characterize individual and interactive scene features to score our scenes and create a distribution shift.

B. Robust Social Navigation

Even though some RL-based models have been proposed for ensuring safety in social settings, they don't extend well to increasingly complex settings. The need to design a reward function based on domain-specific, safety-relevant features constrains them limiting scalability and robustness [14, 15, 16, 17]]. While data-driven approaches can capture the dynamics of social interplay without explicitly hand-crafting a reward function, they are difficult to deploy due to rogue data and covariate drifts [18, 19]. Some papers tackle this by using gradients generated by Q-value functions in Model Predictive Control [20], and [21] extended this method by proposing a generalization to this method using dual control for belief state estimation. In previous work [4] we differ from these approaches by alleviating the need for reward function crafting and simulator training, we use model outputs as action distribution for the downstream planning task handled by an MCTS planner. Recent works have covered MCTS-based robot navigation. In [22] a CNN is utilized to bias the model for online re-planning, however, this work does not extend to the domain of social navigation. [23] proposes a socially-aware planner that utilizes an RNN to simulate future states but does not consider agent-to-agent interaction. Similar to previous work [24] and [25] leverages MCTS and uses a pre-defined reward function to deploy policies. In [SORTS] we rely on expert-based offline policies instead of pre-defined reward functions and simulator training. In this paper, we further extend [4] by improving its performance in out-of-distribution settings by applying a remediation technique to a transformer-based trajectory prediction model that encodes temporal, agent-to-agent, and agent-to-context relationships [3] and integrating it as a social module.

III. Preliminaries

We follow the trajectory prediction formulation in [3] and the social navigation formulation in [4]. Thereafter, we consider the tasks of robust trajectory prediction and robust social navigation under distribution shifts.

We define S as a motion dataset and consider $s \in S$ as a single scenario from this scene pool. Instance s has its distinctive features; in this case, we define it as $\mathbf{X}$, as mentioned in the previous subsection. By scoring these features, a distribution shift can be created. Thus, S can be split into two: S_{ID} for in-distribution scenarios, and S_{OOD} for out-of-distribution settings.

The task of *robust prediction and navigation under distribution shift* is to reduce the decline in performance on safety-relevant metrics, such as loss of separation, when prediction models trained and validated on S_{ID} are tested on S_{OOD}.

IV. Approach

A. Overview

SafeAir, shown in Figure 1, is a safety-oriented framework designed to enhance navigation performance in safety-critical, out-of-distribution scenarios within the aviation domain by leveraging existing datasets and trajectory prediction algorithms. SafeAir uses a data-driven, context-aware trajectory prediction approach and utilizes the previously mentioned distribution shift in its training process. SafeAir also includes a long-horizon planner for navigation. This way both online and offline model enhancement techniques are employed to ensure robustness.

The sub-sections below are organized as follows: Section IV-B describes the scene characterization and scoring approach used for creating a safety-informed dataset split. Next, in Section IV-C we describe the trajectory prediction algorithm and the robustification strategy used for coping with the distribution shift. Similarly, in Section IV-D, we describe the social navigation algorithm that leverages the robustified prediction model to improve downstream performance.

B. Creating a Distribution Shift

We follow the scene characterization and scoring strategy as in [5], which was introduced for the domain of autonomous driving. We adapt it accordingly to the domain of general aviation.

Briefly, the framework first computes low-level, base features within a scenario and then aggregates them to generate a scene score according to safety relevance. The features come in two main categories: (1) *Individual Features*, which characterize each agent's kinematic state, and (2) *Interaction Features*: which characterize agent-to-agent relationships. Table I provides a full list of the features that were considered in each category.

TABLE I: Scenario Characterization Features.

Individual Features	**Interaction Features**
Speed	Time Headway (THW)
Acceleration	Time-to-Collision (TTC)
Jerk	Deceleration Rate to Avoid Crash (DRAC)
Waiting Period at Conflict Point	Minimum Time to Conflict Point (mTTCP)
Speed diff. with Speed Limit	Loss of Separation
Adherence to a Traffic Pattern	Segment-to-Segment Overlap
Trajectory Anomaly	Trajectory-Pair Anomaly

Once the features are computed for a given scene, we linearly combine them to form a final *scene score*, as in [5]. Figure 2 shows the score density function across all scenarios within the `TrajAir` dataset. The figure shows a long tail of highly-scored scenarios, representing the most safety-relevant scenarios. We leverage the final scores to split the dataset into an in-distribution set (for training and validation) and an out-of-distribution set (for testing). The split cut-off is set at the 90% percentile as shown in the figure.

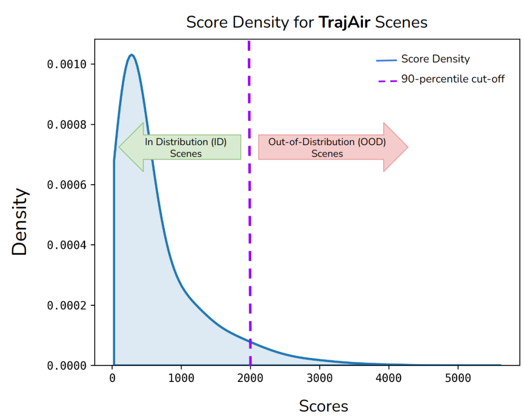

Fig. 2: Score density function across *TrajAir* scenarios.

Finally, Figure 3 shows examples of scenes within the in-distribution vs out-of-distribution sets. For the former, we can observe the scenes represent relatively sparse interactions with agents being sufficiently separated from each other. For the latter, we can observe increasingly more complex scenes, with agents being closer to each other and coordinating multiple simultaneous tasks, *e.g.*, landing, take-off, and cruising.

C. Robust Trajectory Prediction

We use `AmeliaTF` [3] as the backbone for the trajectory prediction task as it was designed for airport surface movement forecasting, making it suitable for our domain. In this section, we briefly cover the details of the model but focus more on the modifications made to adapt it to the domain of general aviation. An overview of our modified design is shown in Figure 4. For in-depth implementation details, we refer the reader to [3, 5].

1) Model Details: `AmeliaTF` is a transformer-based model that layer-wise encodes temporal, agent-to-agent, and agent-to context relationships. The model is composed of three main sub-modules: A *scene representation* module that determines agents of interest (i.e. ego agent); a *scene encoder* that determines the relationship between agent, context, and time; a *trajectory decoder* that models the predictions which corresponding confidence scores come from a Gaussian Mixture Model [26].

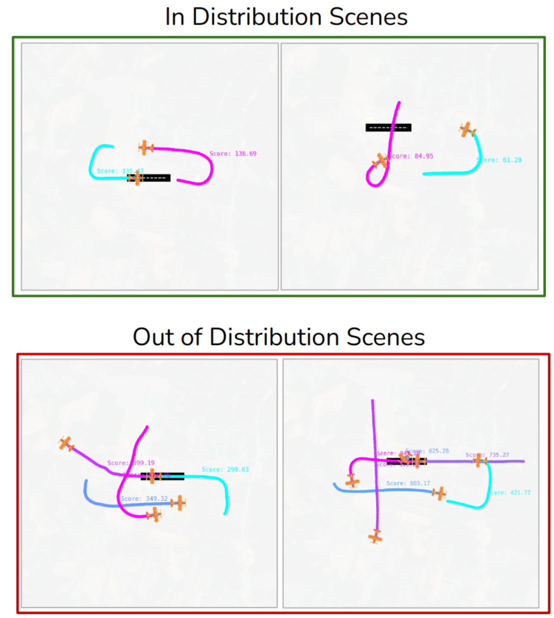

Fig. 3: Examples of in-distribution (top, green) vs out-of-distribution (bottom, red) scenarios using the safety-informed distribution shift strategy.

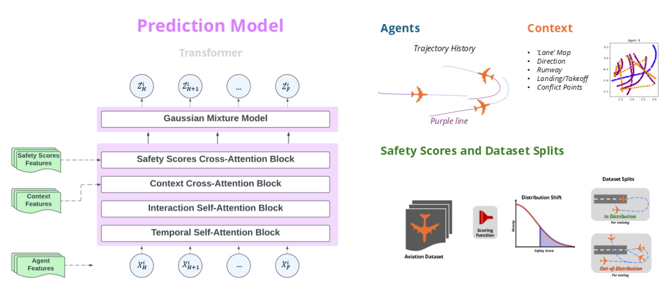

Fig. 4: Overview of `SafeAir`'s trajectory prediction model. It consists of a transformer-based model that encodes temporal, agent-to-agent, and agent-to-map relationships sequentially. To robustify it, we additionally encode agent-to-scores, to learn the relationship between an agent's behavior and its safety score.

A semantic map was encoded to integrate context features specific to the shared airspace domain into the already existing TF architecture. This map was built using expert paths obtained from real pilots. Its attributes are a 'lane' map (i.e. flight patterns), conflict points between those patterns, maneuver as in takeoff or landing, direction (cardinal points), and runway where the maneuver is being made.

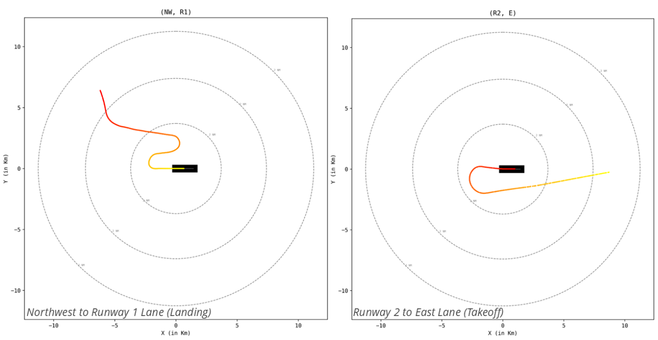

Fig. 5: Figure showing two expert paths, landing (left) and takeoff (right)

2) Robustification: To robustify the model against the OOD scenes obtained in the previous section, we integrate an additional cross-attention block, which encodes the scene scores using an MLP-based feature encoder, and then runs agent-to-score cross-attention. We then apply a collision remediation strategy by conditioning the model's loss function with safety scene scores. This weighted loss, `Wloss` is calculated as follows:

$$\texttt{Wloss} = \frac{1}{B}\sum_{b}^{B} \texttt{SceneScore}^{(b)} \cdot (\texttt{Loss}_{reg} + \texttt{Loss}_{coll})^{(b)}$$

where B is the scene batch and we consider $b \in B$ as a single scene from the batch. $\texttt{Loss}_{reg}$ is a Gaussian negative log-likelihood loss between trajectory predictions and its target (i.e. ground truth predictions) and $\texttt{Loss}_{coll}$ is a cross-entropy loss of the prediction scores with its target being the indexes of the ego agent's predictions with fewer collisions when comparing those predictions to those of other agents. We denote a scene's safety score as $\texttt{Scene-Score}^{(b)}$ and do the same for both losses.

D. Robust Social Navigation

We build upon `SoRTS` [4] for the downstream navigation task. Here, we delve into detail into the adaptations of how the model is adapted for dealing with OOD scenarios. An overview of the model is shown in Figure 6. For implementation details, we refer to [4].

1) Model Details: Centered on performance and safety, our navigation planner is designed to explore multi-modal futures and select the optimal trajectory. It employs a Monte Carlo Tree Search (MCTS) algorithm supported by two key modules: a *Reference Module* that provides global reference paths to guide the agent, and a *Social Module* that manages short-horizon social interactions between agents. The planner integrates inputs from both modules to compute action distributions. Due to the tree structure of MCTS, each node

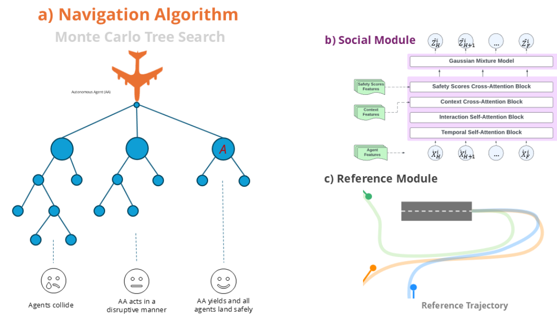

Fig. 6: Overview of `SafeAir`'s navigation algorithm. It builds upon [4], an MCTS-based algorithm that biases its tree search through a socially-aware prediction model. Here, our robustified prediction model is used as the socially-aware prediction model to improve navigation performance in unseen settings.

represents a potential future state, and online collision checks are conducted, with branches leading to collisions being pruned. This approach enables the MCTS model to identify the best collision-free, high-reward trajectory and perform backpropagation, resulting in long-horizon, socially desirable navigation.

2) Robustification: `SoRTS` is originally unequipped to cope with OOD scenarios. Thus, we integrate our robustified prediction model to provide socially-aware and safety-informed decision-making.

V. Experimental Setup

A. Trajectory Prediction

We utilize the `TrajAir` dataset [2] to train our model. As mentioned in Section IV-C, we adapt the dataset to include map context information and improve the overall situational awareness, and we adapt the model to better cope with safety-relevant out-of-distribution scenarios. Thus, in our experiments, we seek to assess the effect of adding context to the model, as well as the robustification effect.

First, in our evaluations, we consider two dataset splits:

- `Uniform`: the original train/val/test random splitting methodology. Experiments `Exp. 1` and `Exp. 2`, explained below, are assessed under this split.
- `SafeShift`: the dataset is split according to Section IV-B. Experiments `Exp. 3` and `Exp. 4`, explained below, are assessed under this split.

1) Context Experiments: The first set of experiments is focused on assessing the performance of the model when integrating map information as context. In Table III we use `Exp. 1`, to refer to the trajectory-only model, and `Exp. 2` as the context-aware model.

2) Robustness Experiments: We then seek to assess the model's performance under the `SafeShift` split. To do so, we perform `Exp. 3`, where we first assess the performance of the context-aware model without the robustification described in Section IV-C. Then, we perform `Exp. 4` to assess the model's performance upon applying the robustification strategies.

3) Metrics: For all experiments explained above, we consider the following metrics:

- *Average Displacement Error (ADE)*: minimum average error across time steps between the predicted trajectories and ground truth (GT).
- *Final Displacement Error (FDE)*: minimum average error for the last time step between the predicted trajectories and the GT.
- *Loss of Separation, Pred-to-GT (LoS-GT)*: Average number of frames an agent's prediction breaches a minimum distance of 0.1km between an agent's prediction and other agents' GT is breached.
- *Loss of Separation, Pred-to-Pred (LoS-Pred)*: Average number of frames an agent's prediction breaches a minimum distance of 0.1km between an agent's prediction and other agents' predictions is breached.

4) Hyperparameters: We run a hyperparameter search to find the best configuration based on ADE/FDE metrics. The final hyperparameters are summarized in Table II. Here, *Polylines*, is the amount of context will an agent take into consideration. It can be considered as its field of view.

TABLE II: Model hyperparameters.

Model	Learning Rate	Batch Size	Embed Size	Polylines
`Exp. 1`	0.0001	64	512	-
`Exp. 2`	0.0001	32	256	300
`Exp. 3`	0.0001	32	256	300
`Exp. 4`	0.0001	32	256	250

B. Social Navigation

The experimental setup for navigation is a landing task. It is designed to assess the coordination of multiple agents trying to land on the same side of a runway in a non-towered airport. In the absence of a central authority for orchestrating maneuvers, social coordination is vital for a safe landing which also has to follow guidelines by the FAA[27]. In these simulations, two to five agents are spawned anywhere around a ten-kilometer radius of the runway. Agents' incoming directions can be all cardinal directions. An agent is considered unsuccessful if it violates the minimum separation distance from another agent, exits the spawn radius, or exceeds the maximum number of allowed steps.

1) Navigation Experiments: We perform the experiments listed below to assess navigation performance with the trajectory prediction model improvements:

- `SoRTS`: baseline algorithm. Its social component was trained using global positioning information and no context.
- `SafeAir(+Exp. 3)`:, we replace the social component with the architecture described in section IV-C, which leverages map information and the model architecture from [3].
- `SafeAir(+Exp. 4)`:[1], we leverage the proposed trajectory prediction model which includes the components from `Exp. 3` and the robustification strategy.

2) Metrics: For the navigation experiments, we consider the following metrics:

- *Success Rate (SR)*. Percentage of successful landings.
- *Loss of Separation Rate (LoS)*. Percentage of collisions amongst agents.
- *Timeout (T/O)*: Timeout. Percentage of instances an agent could not complete the task in time.

VI. Results

A. Trajectory Prediction

Table III summarizes the prediction results across the experiments described in Section V-A.

1) Context Experiments: First, when training in the uniform split, we can see a performance improvement of 13% and 33% in *ADE* and *LoS*, respectively, when adding context to the model. Even though these models are not assessed on the out-of-distribution test bed, the diminished *LoS* rates in `Exp. 2` are a result of better situational awareness. By training this context-aware model (`Exp. 2`) on a safety-informed split, the increased *LoS* rates were an anticipated outcome due to the testing on out-of-distribution scenarios.

TABLE III: Trajectory prediction results under the `Uniform` and `SafeShift` [5] sets. ADE/FDE are in meters. LoS is an average number of frames.

Model	Modalities: Trajectory	Modalities: Context	Modalities: Scores	Dataset Split	ADE / FDE	LoS GT/Pred
`Exp. 1`	✓			`Uniform`	37.3 / 68.4	0.11 / 13.89
`Exp. 2`	✓	✓		`Uniform`	32.2 / 57.2	0.10 / 9.24
`Exp. 3`	✓	✓		`SafeShift`	**27.2 / 48.9**	0.86 / 19.99
`Exp. 4`	✓	✓	✓	`SafeShift`	**32.8 / 59.9**	**0.84 / 6.16**

2) Robustness Experiments: Here, we first show through experiment `Exp. 3` that upon training our model using our safety-informed dataset split, the Loss of Separation metrics worsened significantly by 760% and 116% when compared to other agents' ground truth and predictions, respectively. This trend follows also from [5], and further highlights the need and motivation for robustification strategies that provide the model with the ability to cope with unseen and potentially safety-critical scenarios. Finally, `Exp. 4` shows that our approach to integrating remediation techniques into the prediction model proved effective. Specifically, our robustification strategy achieved a 70% reduction in prediction collision rates. While this improvement came at the cost of 20% drop in *ADE/FDE* performance, a decrease in *LoS-Pred* is significantly more important in the context of safety-criticality.

Figure 7 shows a comparison within the same scenario, it can be noted that Exp. 4 diverges from the agent ahead,

[1]Note: The integration of `Exp. 4` into the navigation model did not perform as expected. Due to time constraints, we weren't able to assess and fix potential errors. Thus, the results of this experiment are currently not reported in the navigation table. We are currently working to resolve it.

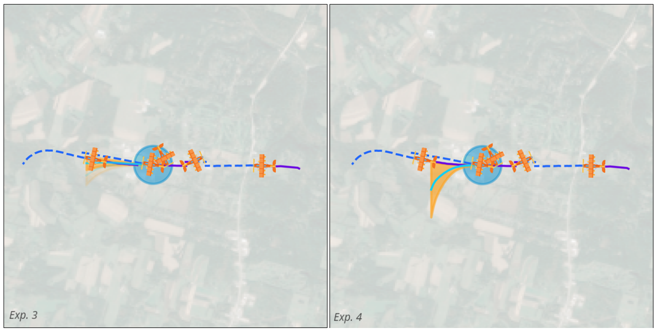

Fig. 7: Trajectory prediction results for `Exp. 3` (left) and `Exp. 4` (right).

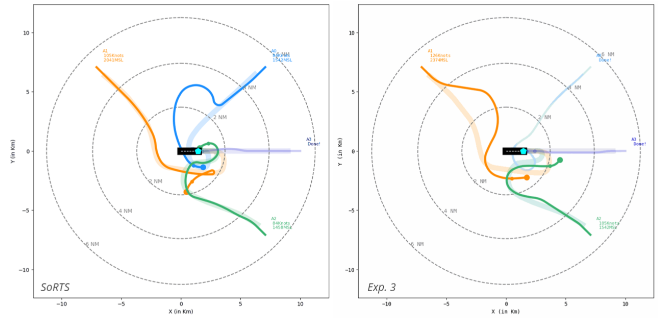

Fig. 8: Navigation results for `SoRTS`(left) and `SafeAir`(`+Exp. 3`) (right). Both snapshots were taken at the same timestep.

attempting to avoid a social conflict. In contrast, Exp. 3 maintains its trajectory, resulting in an uncomfortable situation.

B. Navigation

Table IV illustrates the performance comparison between `SafeAir` `(+Exp.3)` ablation and `SoRTS`. The success rate improves by 4.3% and 13% in scenarios with two to three agents, suggesting that adding domain-specific context to the model helps in the tasks of social-coordinated landing. However, in more out-of-distribution settings including four and five agents, the success rate decreases by 4.7% and 17%, as anticipated. Despite this, the loss of separation rate decreases significantly by 30% and 75% indicating that even as the success rate declines, the system becomes more effective at maintaining safe separation between agents.

TABLE IV: Navigation prediction results of ablations `SoRTS` and `SafeAir` `(+Exp.3)`

Num. Agents	Model	Success Rate (SR)	LoS Rate	T/O
2	SoRTS	93%	4%	1%
	SafeAir(+Exp. 3)	97%	1%	1%
3	SoRTS	86%	13%	0%
	SafeAir(+Exp. 3)	97%	13%	1%
4	SoRTS	86%	13%	1%
	SafeAir(+Exp. 3)	82%	9%	8%
5	SoRTS	82%	16%	2%
	SafeAir(+Exp. 3)	66%	4%	29%

Figure 8 presents a comparison between Experiment 3 and `SoRTS` in the same scene and at the same timestep. It is apparent how the proposed model's action distributions lead to smoother, less erratic flight patterns. This results in more predictable and safer interactions between agents.

VII. Conclusion

To ensure safe and seamless navigation in human-centered domains, such as aviation, it is crucial to address out-of-distribution scenarios, as these unseen settings can be exhibited in the real world and they present unique challenges that differ significantly from model training datasets. This can result in sub-optimal performance and a safety risk for all agents in the scenario. Ensuring robustness is the only way to develop trustworthy models. We aim to explore various remediation strategies to enhance the robustness of our framework. This means searching for the cause of performance deterioration in trajectory prediction to fortify the algorithm's ability to handle safety-critical situations effectively.

Acknowledgments

This work was possible thanks to the constant mentoring of PhD student Ingrid Navarro. Thanks to Prof. Jean Oh and the BIG Lab for their support. Thanks to everyone who makes the RISS program possible, especially Rachel Burcin, Morgan Grimm, and John Dolan.

References

[1] Ingrid Navarro and Jean Oh. "Social-PatteRNN: Socially-Aware Trajectory Prediction Guided by Motion Patterns". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022.

[2] Jay Patrikar et al. "Predicting like a pilot: Dataset and method to predict socially-aware aircraft trajectories in non-towered terminal airspace". In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 2525–2531.

[3] Ingrid Navarro et al. "AmeliaTF: A Large Model and Dataset for Airport Surface Movement Forecasting". In: *AIAA AVIATION FORUM AND ASCEND 2024*. 2024. DOI: 10.2514/6.2024-4251. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2024-4251. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2024-4251.

[4] Ingrid Navarro et al. "SoRTS: Learned Tree Search for Long Horizon Social Robot Navigation". In: *IEEE Robotics and Automation Letters* 9.4 (2024), pp. 3759–3766. DOI: 10.1109/LRA.2024.3370051.

[5] Benjamin Stoler et al. "SafeShift: Safety-Informed Distribution Shifts for Robust Trajectory Prediction in Autonomous Driving". In: *2024 IEEE Intelligent Vehicles Symposium (IV)*. 2024, pp. 1179–1186. DOI: 10.1109/IV55156.2024.10588828.

[6] Erica Weng et al. “Joint Metrics Matter: A Better Standard for Trajectory Forecasting”. In: *CoRR* abs/2305.06292 (2023). DOI: 10.48550/arXiv.2305.06292. arXiv: 2305.06292.

[7] Saeed Saadatnejad et al. “Are socially-aware trajectory prediction models really socially-aware?” In: *Transportation research part C: emerging technologies* 141 (2022), p. 103705.

[8] Qingzhao Zhang et al. “On adversarial robustness of trajectory prediction for autonomous vehicles”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 15159–15168.

[9] Angelos Filos et al. “Can autonomous vehicles identify, recover from, and adapt to distribution shifts?” In: *International Conference on Machine Learning*. PMLR. 2020, pp. 3145–3153.

[10] Masha Itkina and Mykel Kochenderfer. “Interpretable self-aware neural networks for robust trajectory prediction”. In: *Conference on Robot Learning*. PMLR. 2023, pp. 606–617.

[11] Luyao Ye, Zikang Zhou, and Jianping Wang. “Improving the Generalizability of Trajectory Prediction Models with Frenet-Based Domain Normalization”. In: *arXiv preprint arXiv:2305.17965* (2023).

[12] Moritz Werling et al. “Optimal trajectory generation for dynamic street scenarios in a frenet frame”. In: *2010 IEEE international conference on robotics and automation*. IEEE. 2010, pp. 987–993.

[13] Parth Kothari et al. “Motion style transfer: Modular low-rank adaptation for deep motion forecasting”. In: *Conference on Robot Learning*. PMLR. 2023, pp. 774–784.

[14] Sango Matsuzaki and Yuji Hasegawa. “Learning Crowd-Aware Robot Navigation from Challenging Environments via Distributed Deep Reinforcement Learning”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 4730–4736.

[15] Yu Fan Chen et al. “Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning”. In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2017, pp. 285–292.

[16] Changan Chen et al. “Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 6015–6022.

[17] Chieh-En Tsai and Jean Oh. “A generative approach for socially compliant navigation”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 2160–2166.

[18] Mohammad Ali Bashiri, Brian Ziebart, and Xinhua Zhang. “Distributionally Robust Imitation Learning”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 24404–24417.

[19] Felipe Codevilla et al. “Exploring the limitations of behavior cloning for autonomous driving”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9329–9338.

[20] Dorsa Sadigh et al. “Planning for cars that coordinate with people: leveraging effects on human actions for planning and active information gathering over human internal state”. In: *Autonomous Robots* 42.7 (2018).

[21] Haimin Hu and Jaime F Fisac. “Active uncertainty reduction for human-robot interaction: An implicit dual control approach”. In: *International Workshop on the Algorithmic Foundations of Robotics*. Springer. 2022, pp. 385–401.

[22] Julius Rückin, Liren Jin, and Marija Popović. “Adaptive informative path planning using deep reinforcement learning for UAV-based active sensing”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 4473–4479.

[23] Stuart Eiffert et al. “Path planning in dynamic environments using generative rnns and monte carlo tree search”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 10263–10269.

[24] Benjamin Riviere et al. “Neural tree expansion for multi-robot planning in non-cooperative environments”. In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 6868–6875.

[25] Jeongwoo Oh et al. “SCAN: Socially-Aware Navigation Using Monte Carlo Tree Search”. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 7576–7582.

[26] Yuning Chai et al. “Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction”. In: *arXiv preprint arXiv:1910.05449* (2019).

[27] Federal Aviation Administration. *Aeronautical information manual*. 2023.