A Novel Airport Dependent Landing Procedure Based on Real-World Landing Trajectories

Ensieh Alipour * and Seyed Mohammad-Bagher Malaek

Posted Date: 11 February 2026

doi: 10.20944/preprints202602.0823.v1

Keywords: adversarial inverse reinforcement Learning (AIRL); proximal policy optimization (PPO); aircraft; imitation learning; landing procedure; data-driven framework; aviation

 doi:10.20944/preprints202602.0823.v1

Article

A Novel Airport Dependent Landing Procedure Based on Real-World Landing Trajectories

Ensieh Alipour [1,*] and Seyed Mohammad-Bagher Malaek [2]

[1] PhD Candidate, Aerospace Eng. Dept., Sharif University of Technology

[2] Professor at Aerospace Eng. Dept., Sharif University of Technology

* Correspondence: en.alipour@gmail.com or alipour.ensieh@ae.sharif.edu

Abstract

This study presents a novel data driven framework for developing airport-specific landing policies and procedures from historical successful landing data. The proposed process, termed the Airport Dependent Landing Procedure (ADLP), is motivated by the fact that airports rely on uniquely tailored approach charts reflecting local operational constraints and environmental conditions. While existing approach charts and landing procedures are primarily designed based on expert knowledge, safety margins, and regulatory conventions, the authors argue that data science and data mining techniques offer a complementary and empirically grounded methodology for extracting operationally meaningful structures directly from historical landing data. In this work, we construct a probabilistic three-dimensional environment from real-world aircraft approach trajectories, capturing spatiotemporal relationships under varying atmospheric conditions during approach. The proposed methodology integrates Adversarial Inverse Reinforcement Learning (AIRL) with Recurrent Proximal Policy Optimization (R-PPO) to establish a foundation for automated landing without pilot intervention. AIRL infers reward functions that are consistent with behaviors exhibited in prior successful landings. Subsequently, R-PPO is employed to learn control policies that satisfy safety constraints related to airspeed, sink rate, and runway alignment. Application of the proposed framework to real approach trajectories at Guam International Airport demonstrates the efficiency and effectiveness of the methodology.

Keywords: adversarial Inverse reinforcement learning (AIRL); proximal policy optimization (PPO); aircraft; imitation learning; landing procedure; data-driven framework; aviation

1. Introduction

The landing phase remains one of the most demanding and safety-critical segments of all commercial flight operations. It requires precise, continuous management of descent rate, airspeed, lateral alignment as well as energy state amid variable wind conditions. Other parameters, such as traffic constraints and airport geographical constraints usually add to complexities involved. Despite significant advances in autopilot and flight-management systems, as well as "Instrument Landing Systems (ILS)", human pilots still bear primary responsibility for the final approach and landing, where even small deviations can lead to abortion of the process and execution of "go-arounds", or otherwise, runway excursions. On the other hand, with steadily increasing global air traffic and growing interest in higher levels of flight safety with even "single-pilot" operations as well as "crew-less" cargo aircraft—there is a pressing need for a sound a scientific approach to develop landing procedure and policies that could lead to control systems capable of replicating robust, expert-consistent landing performance directly from history of operational evidences at a given airport.

Traditional aircraft control-system design relies heavily on aircraft analytical dynamic models. The process continues with linearized equations of motion and gain-scheduled controllers tuned for different aircraft conditions. Surprisingly enough, such tradition completely ignores the geographical

 doi:10.20944/preprints202602.0823.v1

airport characteristics and solely concentrates on the aircraft and its dynamical characteristics. Not to mention that such labor-intensive approach would not even guarantee any safety measures and still calls for many hours of flight-testing with different weights in different atmospheric disturbances together with specifically tailored trainings at different airports. However, with new advances in Reinforcement learning (RL) techniques, we are at a turning point in the development of new class of control systems that have the ability to adjust themselves for different airport destinations. In this line of thought, in addition to the aircraft, the airport characteristics as well as its approach charts come into the process and play their roles from the beginning. Moreover, there are still no scientific process that any proposed approach chart for any airport is the best-one possible. In fact, as long as such approach charts seem doable for ordinary pilots, they are approved. In this mechanism, it is the novice pilot that has to learn how to land at a given airport and the burden is solely on the pilots to learn how to land at different airports. Nevertheless, the current work aims to enhance the situation and to lower pilot workload, while maintaining accepted level of safety. The idea is simple, with the help of data associated with successful previous landings, it is the aircraft that learns which trajectory to follow and how to land, not the pilot. Following sections descries the process.

2. Learning Mechanisms of Learning Capable Aircraft

In this work, we use Inverse Reinforcement Learning (IRL), and particularly its adversarial variant (AIRL), to address existing challenges to extract landing policies directly from previous successful landing data, instead of "Airport Approach Charts". The process recovers suitable reward functions directly from previous expert demonstrations, eliminating the need for manual reward manipulations. When combined with modern policy optimization methods, such as Proximal Policy Optimization (PPO), AIRL enables the training of policies that closely mimic previous experts' behavior. It has to be noted that unlike most prior works that rely on simulated environments for learning, we explicitly use actual aircraft trajectories so there are neither simplified dynamics involved nor any simplifications that might compromise the safety of the aircraft and its passengers.

The rest of the manuscript describes the framework of the so called "**P**olicy **I**nference from **L**ogged **O**perational **T**rajectories (**PILOT**)" in three sections, as the following:

1. The first section describes how we can collect and prepare the relevant data for a specific flight case, such as ADS-B data during approach to a specific airport.
2. The second section describes the learning process to form a robust, expert-consistent, 3D controller exclusively from real-world data associated with section (1)
3. The third section, describes how the process could be fine-tuned for a specific airport, which in this study is International Airport (PGUM).

With proper data (For this work 1,039 successful approaches), we construct a probabilistic Markov Decision Process (MDP) that captures empirical spatiotemporal relationships in properly selected aircraft states. Different case-studies reveal that during approach phase of the flight, aircraft altitude, lateral deviation, and energy state are minimum necessary. Of course, other parameters could be added to the list, as needed. AIRL process is then applied to infer a reward function consistent with observed aircraft behavioral states, followed by recurrent PPO (R-PPO) training of an LSTM-based policy that aims to produce stable coherent command for the autopilot during approach.

In this approach, we are using all the data associated with "Successful flight conditions" we, obviously, expect to safety and realism to be naturally enforced in all critically selected states. The primary contributions of this work are:

- A new probabilistic model based on real-world data to extract policies without mathematical models.
- Extraction of applied reward functions directly from history of operation to feed into R-PPO, eliminating the need for trial-error in the process of reward adjustments.
- Provision of implicit robustness and stability based on experts in the field historical data.

 doi:10.20944/preprints202602.0823.v1

- Provision of built-in validation process into the model with robustness to initial-state variations, and guarantees valid terminal conditions.

Based on the mentioned contributions, this work paves the way to develop a new class of control system that has the ability to mimic landing policies suitable for both manned and unmanned aircraft systems.

3. Literature Review

Adversarial Inverse Reinforcement Learning (AIRL) has become a central method for extracting reward functions from expert demonstrations using an adversarial discriminator–generator structure. The foundational work of Fu et al. [1] introduced AIRL as a robust alternative to traditional IRL, demonstrating improved reward transferability and policy generalization in continuous-control domains. This triggered a series of studies aimed at improving reward stability, scalability, and sample efficiency.

Several methodological extensions build on this foundation. Model-based variants such as MAIRL [2] introduced a self-attention dynamics model to reduce optimization variance, while option-based approaches such as oIRL [3] improved reward disentanglement through temporally extended actions. Off-policy AIRL [4] addressed the high sample cost of the original on-policy formulation, and generative extensions combining AIRL with DDPG [5] reduced training time through deterministic sampling. Hybrid AIRL (HAIRL) [6] incorporated curiosity-driven exploration to mitigate AIRL's tendency toward premature exploration saturation.

A major application area for AIRL is sequential decision-making. In autonomous driving, several augmented AIRL frameworks [7–9] incorporated semantic features to improve learning stability and decision accuracy in interactive traffic settings, consistently outperforming GAIL and reinforcement-learning baselines. Beyond single-agent domains, multi-agent and adversarial extensions such as MA-AIRL [10] and I-IRL [11] expanded AIRL to strategy recovery in Markov games and adversarial environments. More recent work has addressed long-horizon challenges: hierarchical critics and subtask decompositions in H-AIRL [12] and SC-AIRL [13] improved credit assignment and exploration in complex robotic manipulation problems.

Overall, these studies highlight AIRL's strengths in reward interpretability, robustness, and behavioral fidelity. However, the existing literature exhibits several limitations. AIRL research remains heavily dependent on simulated environments and rarely incorporates high-dimensional, physics-informed state representations.

Building on the advancements in Adversarial Inverse Reinforcement Learning (AIRL) for robust reward recovery from expert demonstrations, broader applications of Deep Reinforcement Learning (DRL) and Deep Imitation Learning (DIL) have extended these techniques to safety-critical domains.

Deep Reinforcement Learning (DRL) and Deep Imitation Learning (DIL) have emerged as leading paradigms for autonomous control in complex, safety-critical domains such as autonomous driving and mobile robotics [14–17]. These approaches have been successfully extended to Unmanned Aerial Vehicles (UAVs), enabling high-performance autonomous flight—including human-champion-level performance in real-world drone racing [18,19]. Notable applications include continuous-action obstacle avoidance [20], wind-aware navigation procedure optimization [14], and airport traffic management [21]. Algorithms such as Proximal Policy Optimization (PPO), Deep Deterministic Policy Gradient (DDPG), and modular or curriculum-based training have proven effective for motion planning and control in dynamic, uncertain environments [18,22,23].

Despite these advances, the "sim-to-real" gap remains a fundamental obstacle: most DRL methods perform well in simulation but degrade significantly in physical deployment due to unmodeled dynamics, sensor noise, and environmental variability [24,25]. Domain randomization and high-fidelity simulators offer partial solutions [24], yet direct learning from real-world data is increasingly recognized as a more robust path forward. Offline Reinforcement Learning, which trains solely on static datasets without online interaction, has gained prominence for applications where real-time exploration is expensive or hazardous [26].

 doi:10.20944/preprints202602.0823.v1

A parallel challenge in applying RL to aircraft landing is the difficulty of manually designing reward functions that capture the subtle, multi-objective trade-offs expert pilots employ. Inverse Reinforcement Learning (IRL), particularly Adversarial IRL (AIRL), resolves this by inferring reward functions directly from expert demonstrations, thereby producing human-like policies without explicit reward engineering [27]. Combining AIRL with recurrent policy optimization (e.g., PPO with LSTM architectures) further supports stable, long-horizon decision-making required for extended flight segments.

Safety and certifiability are non-negotiable in aviation. Recent surveys stress the need for constrained learning, formal safety guarantees, and explicit enforcement of operational envelopes [20,28]. Terminal constraints on critical variables (airspeed, sink rate, alignment) and rigorous containment analysis within observed state distributions directly address these requirements.

The PILOT framework advances this body of work by learning robust, pilot-like landing policies exclusively from 1,039 real-world ADS-B approach trajectories collected at Antonio B. Won Pat International Airport (PGUM), Guam, across diverse meteorological conditions and aircraft types. By constructing an empirical probabilistic Markov Decision Process, employing AIRL to recover pilot-consistent rewards, and training a recurrent PPO policy with explicit safety constraints, PILOT eliminates both hand-crafted rewards and dependence on analytic flight-dynamics models. Validation demonstrates close statistical agreement with expert trajectories and physical plausibility up to the vicinity of the runway. This fully offline, data-anchored approach directly mitigates the sim-to-real gap and establishes the feasibility of deriving versatile, expert-level guidance for both manned and future unmanned aircraft using only operational surveillance data.

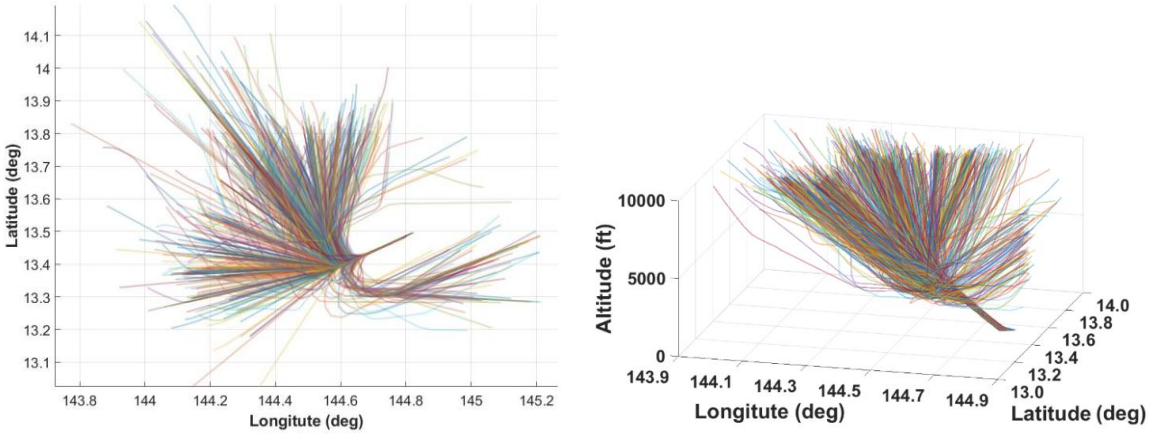
4. Dataset and Preprocessing

The flight dataset used in this study [29] focuses on approach and landing operations at Guam International Airport (PGUM). The details of the dataset are shown in Table 1. A total of 1,039 flight trajectories from 142 individual aircraft were collected, covering twelve major aircraft types, including the Airbus A321 and A330, as well as the Boeing 737, 757, 767, 777, and 787 series. All trajectories are related to the PGUM; therefore, the present study evaluates generalization across aircraft types within a single-airport operational context, not across airport geometries or procedures. These flights represent a diverse range of operational conditions and configurations typical of commercial approach procedures. The corresponding 2D and 3D views are shown in Figure 1.

Table 1. Flight Dataset Summary for Guam International Airport (PGUM).

No.	Airplane Type	Number of Flights	Number of Airplanes of this Type
1	Airbus A321-271N (A21N)	56	15
2	Boeing 767-346-ER (B763)	91	9
3	Boeing 737-8K5 (B738)	384	34
4	Airbus A330-200 (A332)	1	1
5	Airbus A330-322 (A333)	18	10
6	Boeing 777-2B5-ER (B772)	31	5
7	Boeing 777-3B5 (B773)	66	4
8	Boeing 777-3B5-ER (B77W)	91	33
9	Boeing 787-9 Dreamliner (B789)	22	8
10	Boeing 757-230(PCF) (B752)	64	3
11	Airbus A321-231 (A321)	109	19
12	Boeing 737 MAX 8 (B38M)	106	2
Total		**1039**	**142**

 doi:10.20944/preprints202602.0823.v1



(**a**) (**b**)

Figure 1. Approaches to Guam Airport Located at 13.48° N, 144.80° E: (**a**) 2D view; (**b**) 3D view.

Each trajectory contains 1-Hz measurements recorded during the terminal descent phase. All records were filtered to remove incomplete, duplicated, or physically inconsistent samples. Minor temporal gaps were filled using linear interpolation. The resulting dataset provides the foundation for constructing the probabilistic flight environment.

Meteorological variability during the 2025 data-collection period at Guam International Airport (PGUM) was characterized using routine aerodrome weather reports (METAR) [30]. METAR observations for PGUM were retrieved from the Iowa Environmental Mesonet (IEM) ASOS/AWOS/METAR archive using the asos.py interface [31]. The METAR wind fields (direction and speed) were used to compute the crosswind component relative to RWY 06L, and temperature/dew-point observations were used to compute relative humidity [30]. For visualization, months were grouped into a dry season (January-June) and a rainy season (July-December), consistent with climatological descriptions for Guam [32]. Figure 2 provides a compact summary for 2025, including monthly precipitation totals, the monthly distribution (P10 - P50 - P90) of the RWY 06L crosswind component, and monthly median relative humidity. The dashed 16 kt line in the crosswind panel denotes the FAA airport design allowable crosswind component used in runway wind coverage analyses, included here as a design reference rather than an aircraft operational landing limit [33].

 doi:10.20944/preprints202602.0823.v1

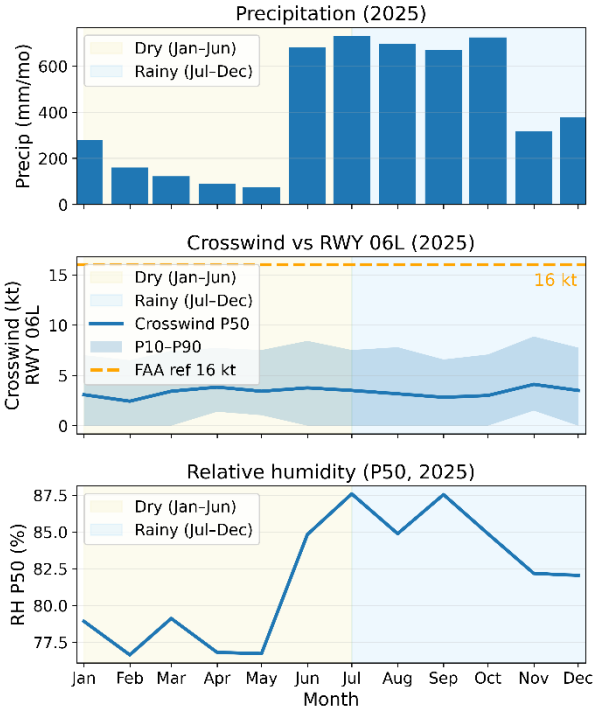


Figure 2. PGUM meteorological summary for 2025 (RWY 06L). Monthly precipitation totals, RWY 06L crosswind component (P10 - P50 - P90), and monthly median relative humidity for PGUM in 2025, with dry (Jan-Jun) and rainy (Jul-Dec) seasons indicated.

5. State–Action–Environment Modeling

5.1. State Representation

The aircraft state is modeled using a nine-dimensional vector containing geometric, kinematic, and energetic quantities relevant to approach and landing:

$$s = [D, XTE, H, V, V_z, G/S, \psi_{err}, \dot{\psi}, h_e] \quad (1)$$

These features represent distance and alignment relative to the runway, vertical profile, heading behavior, and total energy state. A detailed list of all variables, and their physical meaning is provided in Table 2.

Table 2. State variables and final discretization widths.

Symbol	Description	Unit	BIN Width (h)
D	Distance to touchdown	NM	0.05
XTE	Cross-track error	NM	0.015
H	Altitude	ft	5
V	Airspeed	kt	2
V_z	Vertical speed	fpm	50
G/S	Glide Slope	ft/NM	10
ψ_{err}	Heading error	deg	1
$\dot{\psi}$	Heading rate	deg/s	0.1
h_e	Specific energy height, $h_e = h + V^2/2g$	ft	70

5.2. Robust Normalization and Grid Construction

To ensure consistent scaling and avoid the influence of outliers, robust lower and upper bounds for each state variable were extracted directly from the flight dataset using quantile-based clipping. These bounds define the operational envelope used throughout the learning process.

 doi:10.20944/preprints202602.0823.v1

A uniform discretization grid was then constructed: a single bin width per dimension was computed using the Freedman-Diaconis [34] rule and constrained within physically meaningful limits (e.g., minimum altitude resolution, maximum allowable slope step).

The bin width h is defined as:

$$h = 2\,\frac{IQR(x)}{n^{1/3}} \tag{2}$$

where IQR(x) is the interquartile range of the data and n is the sample size. This rule provides a data-adaptive, outlier-robust discretization scale, ensuring the grid is neither too coarse nor too fine before applying domain-specific constraints.

These bin widths were held constant across the full domain and are summarized in Table 2.

This procedure yields a stable and interpretable partition of the continuous state space, forming the basis for constructing the empirical transition model and extracting expert demonstrations.

5.3. Demonstration Extraction

The flight dataset was split into training, validation, and test subsets. Each trajectory was converted into a sequence of expert demonstrations by labeling pilot control commands (actions) vertical (climb/level/descend) and lateral (left/straight/right) based on changes in altitude and heading rate. These labeled demonstrations serve as the supervisory signal for reward learning and policy optimization.

5.4. Action Space and Expert Labeling

The control space is discretized into nine primitive actions, representing the Cartesian combination of vertical and lateral intentions:

$$\mathcal{A} = \{climb, level, descend\} \times \{left, straight, right\} \tag{3}$$

Actions are labeled from flight data by analyzing local altitude and heading changes. A dynamic vertical threshold

$$\varepsilon_H = max\,(0.25 \times H_{bin}, 2\,ft) \tag{4}$$

and a turn threshold, $\varepsilon_\psi = 2^\circ$, were used to classify each time step into one of the nine action categories. This results in discrete state–action pairs (s, a) suitable for transition modeling.

6. Training Methodology

6.1. Environment and Demonstration Preparation

The training environment is a data-derived Markov decision process formed from two empirical components: the transition kernel $P[s, a \to s']$, obtained from state–action frequency counts, and the conditional feature statistics $\mu(s, a \to s')$, which summarize the continuous flight variables associated with each discrete transition. States correspond to quantized combinations of flight parameters, and next state samples are drawn directly from observed transitions in the dataset. Expert demonstrations extracted from 1 Hz flight recordings provide sequences of 9-dimensional states s_t and discrete pilot actions a_t, formatted for the AIRL algorithm. The environment itself is reward-free; the learned reward $R_\theta(s, a, s')$ is applied later through a wrapper that evaluates it online using the trained reward network.

6.2. Learning Framework and Model Architecture

The proposed model is trained using a unified two-stage learning framework that integrates Adversarial Inverse Reinforcement Learning (AIRL) for reward inference with a Recurrent Proximal Policy Optimization (R-PPO) agent for policy optimization. This architecture enables the system to first recover a latent reward function directly from expert flight demonstrations and subsequently

 doi:10.20944/preprints202602.0823.v1

learn a stable closed-loop control policy capable of reproducing pilot-like behavior across extended approach segments. In the first stage, AIRL is used to infer an intrinsic reward signal $R_\theta(s, a, s')$ that best explains the demonstrated trajectories. AIRL formulates the problem as an adversarial game between a discriminator $D_\theta(s, a, s')$, which attempts to distinguish expert transitions from those generated by a provisional policy, and a generator that represents a PPO agent acting within a data-driven Markov decision process. Following the original AIRL formulation by [35], the discriminator models the density ratio between expert and policy-generated transitions via:

$$\varepsilon_H = max\ (0.25 \times H_{bin}, 2\,ft) \tag{5}$$

where f_θ is an energy-based reward-shaping function. The intrinsic reward is then recovered through the canonical AIRL transformation:

$$\varepsilon_H = max\ (0.25 \times H_{bin}, 2\,ft) \tag{6}$$

which yields a scalar reward consistent with maximum-entropy inverse reinforcement learning and is theoretically guaranteed to recover a reward that is invariant under potential-based shaping, thereby preserving the optimal policy [36]. The discriminator is implemented as a multilayer perceptron that receives the concatenated transition vector $[s_t, a_t, s_{t+1}]$, normalized using running statistics to ensure numerical stability. The network consists of three fully connected layers with 256, 128, and 64 units with Leaky-ReLU nonlinearities, followed by a sigmoid output that provides the probability of a transition being expert-generated. During AIRL training, the discriminator and generator are optimized alternately: the discriminator is updated to improve its ability to classify transitions, while the PPO generator is updated to maximize the inferred reward function. We used discriminator indistinguishability (accuracy near 0.5) as a heuristic convergence signal, and additionally monitored policy performance metrics to avoid premature stopping due to discriminator underfitting. Figure 3 illustrates the AIRL stage.

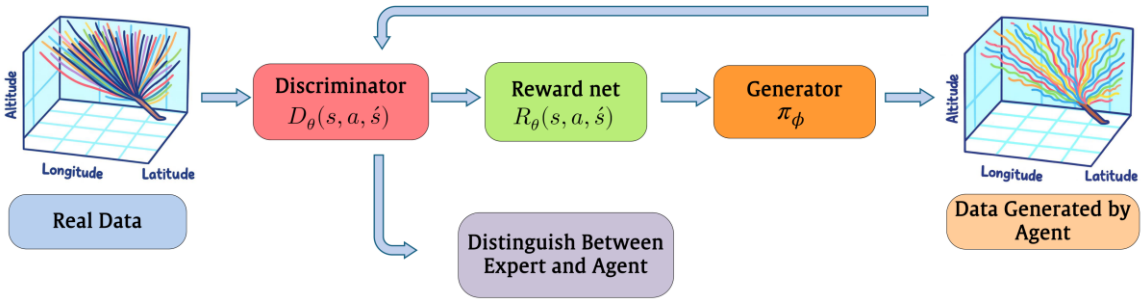


Figure 3. AIRL Stage.

After convergence of the AIRL stage, the learned reward function is fixed and used to train the final controller through Recurrent PPO. The policy network adopts an MlpLstm architecture composed of two 128-unit feed-forward layers followed by a 256-unit LSTM module. The actor head predicts logits over nine discrete control primitives representing vertical and lateral intent classes (climb/level/descend × left/straight/right) inferred from observed state changes., while the critic estimates the value function $V_\phi(s)$. The temporal recurrence in the LSTM enables the agent to integrate sequential information, capture flow-dependent structure in aircraft dynamics, and maintain stable control behavior throughout the approach path. Policy optimization follows the clipped surrogate objective introduced by [37],

$$L^{CLIP}(\pi) = \mathbb{E}[min\ (r_t A_t,\ clip(r_t, 1-\epsilon, 1+\epsilon)A_t)] \tag{7}$$

where

 doi:10.20944/preprints202602.0823.v1

$$r_t = \frac{\pi(a_t \mid s_t)}{\pi_{old}(a_t \mid s_t)} \tag{8}$$

is the probability ratio and A_t denotes the advantage estimate.

Both AIRL and PPO are trained within a data-derived environment constructed from empirical flight trajectories. The continuous state space is discretized using a distribution-adaptive scheme. This rule provides a statistically robust, outlier-tolerant discretization that is subsequently constrained by flight dynamics specific limits such as minimum altitude resolution and maximum allowable slope variation. For each dimension in the state representation, empirical transition counts are accumulated to construct a tabular transition model $P(s' \mid s, a)$, ensuring that the environment captures the stochastic variations present in expert behavior. During both AIRL and PPO training, the agent interacts exclusively with this data-driven MDP, guaranteeing that policy learning remains grounded in the distributional structure of real trajectories rather than relying on analytically simplified models.

This integrated framework allows the controller to learn from the implicit decision-making patterns encoded in pilot demonstrations, recover a reward function that reflects expert intent, and synthesize a recurrent policy that generalizes this behavior across the full landing approach envelope.

The principal hyperparameters used in both training stages are summarized in Table 3. The entire framework of this project is presented in Figure 4.

To promote stable learning, a curriculum mechanism controlled the initial-state distribution and success criteria. Early in training, episodes began close to touchdown (≤ 5 NM) with a wide heading tolerance ($|\psi_{err}| \leq 20°$). As training progressed, the curriculum gradually extended the start distribution to full-approach distances (up to 15 NM) while tightening the success tolerance to 15° and eventually 10°.

Table 3. Training configuration for AIRL and R-PPO stages.

Parameter	AIRL (Reward Inference)	R-PPO (Policy Optimization)	Description
Learning rate	3×10^{-4}	3×10^{-4}	Adam optimizer learning rate
n _steps (per update)	2048	2048	Number of environment steps before each PPO update
Batch size	64	64	Minibatch size for gradient updates
PPO epochs	5	10	Gradient passes per batch
Discount factor (γ)	0.995	0.99	Temporal discounting constant
GAE λ	0.97	0.95	Generalized Advantage Estimation factor
Target KL	0.02	0.03	Early-stop threshold on policy KL (Kullback–Leibler divergence)
Entropy coefficient	0.01	0.01	Exploration regularization term
Max gradient norm	0.5	0.5	Gradient clipping threshold
Discriminator updates per round	2	—	AIRL discriminator optimization steps

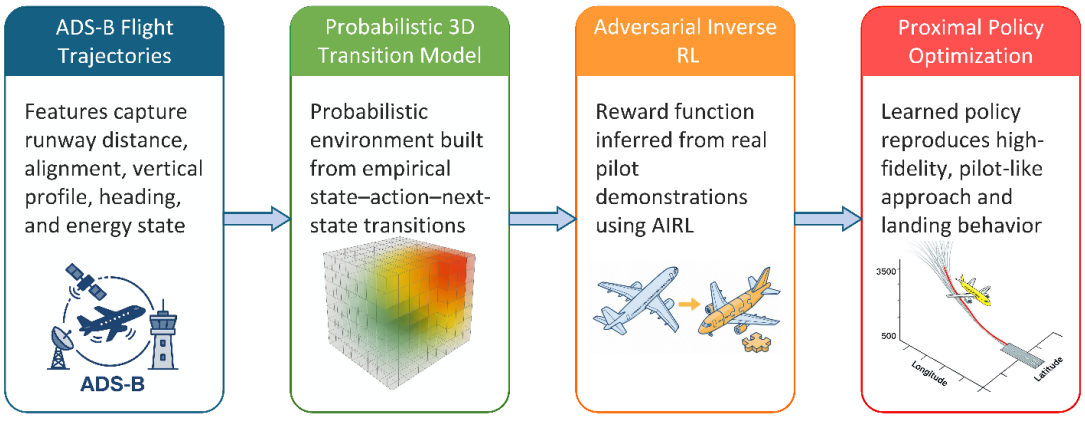


Figure 4. PILOT (Policy Inference from Logged Operational Trajectories) framework for Robust Aircraft Approach Guidance.

 doi:10.20944/preprints202602.0823.v1

This scheduling was implemented using a linear annealing function:

$$r_{\text{near}}(t) = (1 - t)r_0 + tr_1 \tag{9}$$

where r_0 and r_1 denote the near-start ratios at the beginning and end of training. This approach allows the policy to master the final approach region before confronting the full approach envelope.

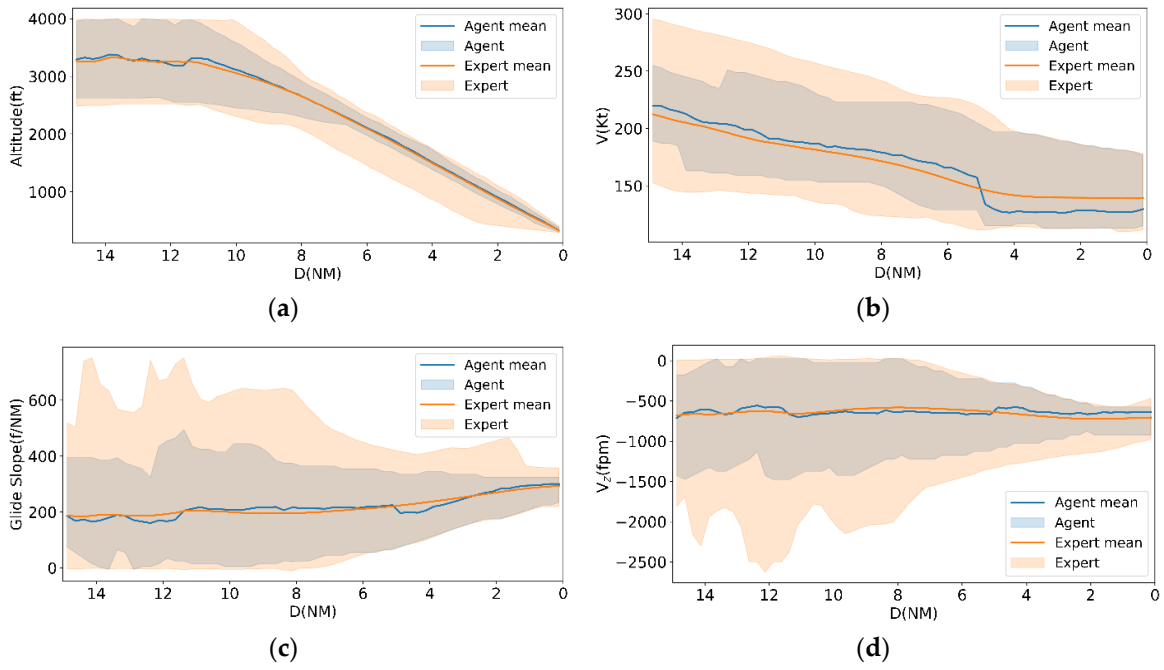
7. Validation and Evaluation

We implement the proposed PILOT framework to learn a landing policy that replicates expert approach behavior. The policy is trained on recorded expert trajectories and evaluated using Monte Carlo rollouts initialized from randomly sampled start states drawn from the dataset. Each rollout is simulated until near touchdown or until a safety-constraint violation occurs. To enable point-wise comparison with expert behavior, all trajectories are re-parameterized by distance to the runway threshold D (ranging from 15 NM to 0 NM), allowing for a direct, point-by-point comparison with expert behavior. We report results from 500 agent rollouts and compare them against 1039 expert approaches. For each variable of interest, we compute the mean profile and the corresponding range across rollouts for both the agent and the expert, assessing agreement in terms of both central tendency and dispersion.

Computations were performed on a standard workstation (Intel Core i7 CPU, 16 GB RAM, single NVIDIA GPU), demonstrating that the PILOT framework can be trained reliably without the need for specialized hardware.

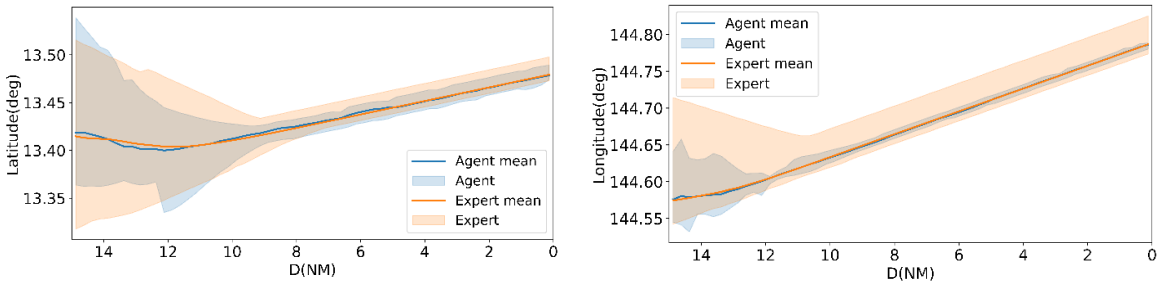
We validate the learned policy against expert trajectories using profile matching over distance and operational speed plausibility at landing. We report these comparisons for:

- Altitude (Figure 5a)
- Speed V (Figure 5b)
- Glide-slope (Figure 5c)
- Vertical speed V_z (Figure 5d)
- Lateral path components (latitude and longitude; Figure 5e-f)



(a) (b) (c) (d)

 doi:10.20944/preprints202602.0823.v1



(**e**) (**f**)

Figure 5. Comparison of Agent and Expert approach profiles versus distance to runway threshold D (NM): (**a**) altitude (ft); (**b**) speed (Kt); (**c**) glide slope (ft/NM); and (**d**) vertical speed V_z (fpm); (**e**) latitude (deg); (**f**) longitude (deg). Solid lines show the mean, and shaded bands indicate the range.

This analysis tests whether the agent matches not only the average behavior but also the dispersion observed in real expert demonstrations. In Figure 5a, the agent's mean altitude profile closely overlaps the expert's mean from 15 NM to touchdown, with strong agreement also in the percentile bands. This indicates that the learned policy reliably reproduces the global descent geometry of the approach.

In Figure 5b-c, the agent generally follows the expert trend. Figure 5d shows that both the agent and expert maintain a broadly consistent descent rate on average. However, the agent exhibits larger short-range fluctuations at some distances.

Latitude and longitude comparisons (Figure 5e-f) show strong overlap between agent and expert means, with narrowing dispersion as $D \to 0$. This is a positive result, suggesting that the policy is not only reproducing the vertical profile correctly but also aligning laterally.

In support of the landing-quality assessment, we verify that the learned policy produces physically plausible approach and touchdown speeds consistent with standard landing envelopes. Table 4 reports representative landing reference speeds V_{ref} (at MLW) and estimated stall speeds V_{s0} for the aircraft types present in the dataset. These values contextualize the touchdown-speed distributions observed in both expert trajectories and agent rollouts and provide a sanity check that the learned policy operates within expected landing-speed regimes. Table 4 also highlights that the expert demonstrations span a wider operating range across aircraft types.

Table 4. Representative landing reference speed V_{ref} (at MLW) and estimated stall speed V_{s0} for aircraft types in the dataset.

No.	Airplane Type (ICAO Code)	V_{ref} @ MLW (kt IAS) [38]	V_{s0} (estimated) (kt IAS) [39]
1	Airbus A321-271N (A21N)	136	104.6
2	Boeing 767-346-ER (B763)	140	107.7
3	Boeing 737-8K5 (B738)	144	110.8
4	Airbus A330-200 (A332)	136	104.6
5	Airbus A330-322 (A333)	137	105.4
6	Boeing 777-2B5-ER (B772)	140	107.7
7	Boeing 777-3B5 (B773)	149	114.6
8	Boeing 777-3B5-ER (B77W)	149	114.6
9	Boeing 787-9 Dreamliner (B789)	144	110.8
10	Boeing 757-230(PCF) (B752)	137	105.4
11	Airbus A321-231 (A321)	142	109.2
12	Boeing 737 MAX 8 (B38M)	145	111.5

To complement the scalar profile comparisons in Figure 5; we provide Supplementary Animation S1 (Figure 6), a synchronized 3D replay of expert and agent approaches. The animation is included as a qualitative diagnostic to (i) reveal transient corrections and short-horizon deviations

 doi:10.20944/preprints202602.0823.v1

that may be partially obscured in percentile bands, and (ii) enable visual consistency checks between 3D path geometry and the corresponding scalar profiles (altitude, descent rate, and speed).

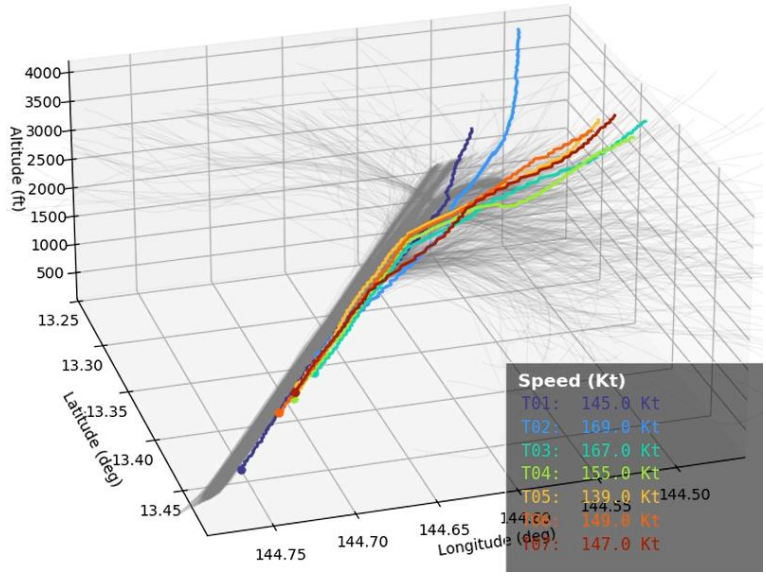


Figure 6. Expert trajectories (gray) overlaid with randomly sampled trajectories generated by the learned policy (from Supplementary Animation S1).

Overall, the combined evidence suggests that the learned policy captures key aspects of expert approach behavior while remaining within physically observed operating regimes. At the same time, residual mismatch in variables closely tied to touchdown quality indicates clear opportunities for improvement within the current data regime. Future work will focus on (i) improving representation and feature design for the approach phase (e.g., better normalization, distance-conditioned features, and more informative terminal descriptors) to reduce short-range oscillations and better match descent-rate behavior near the runway threshold, and (ii) extending the dataset with higher-fidelity near-ground measurements to explicitly model the flare phase, which is not reliably observable from the current ADS-B–based trajectories.

8. Discussion and Future Works

The results demonstrate that the PILOT framework effectively learns robust, pilot-consistent landing policies directly from operational ADS-B trajectories. Across 1,039 real-world approaches at Guam International Airport (PGUM), the learned policy reproduces vertical, lateral, and energy profiles of expert pilots with high fidelity while maintaining physically plausible terminal conditions. Mean profiles and variability ranges of key variables; including altitude, airspeed, glide slope, vertical speed, and lateral alignment; closely match expert demonstrations. These findings highlight the ability of data-driven methods to capture implicit pilot decision-making patterns without relying on analytic flight-dynamics models or hand-crafted reward functions. Future research will focus on enhancing state representation and feature engineering, including distance-conditioned features, improved normalization, and more informative terminal descriptors. Expanding the dataset to incorporate higher-fidelity near-ground measurements would allow more accurate modeling of flare and touchdown phases, which are currently underrepresented in ADS-B–based trajectories. The framework also provides a clear path to reducing human-imposed procedural constraints, particularly reliance on Minimum Descent Altitudes (MDA). By learning landing policies that respect safety limits and pilot-consistent behavior, the system can support continuous-descent operations and adaptive approach paths without strictly defined MDA restrictions, given proper real-time monitoring and safety verification. This represents a potential paradigm shift in approach design,

 doi:10.20944/preprints202602.0823.v1

enabling airport-specific landing procedures to evolve from historical operational evidence rather than prescriptive altitude minima. From a procedural standpoint, a data-driven framework can define conditional, probabilistic minima that reflect actual landing success statistics under well-characterized states of aircraft energy, alignment, and environmental conditions. While the study does not propose immediate replacement of certified minima, airport-specific, data-driven minima could be introduced incrementally as advisory or decision-support layers. Over time, these models may provide a basis for a gradual transition from conservative, rule-based minima toward adaptive minima grounded in operational performance, reducing unnecessary go-arounds, lowering pilot workload, and enabling more consistent approach outcomes—especially at airports with complex terrain or challenging meteorological conditions. Finally, extending this approach across multiple airports and aircraft types, combined with formal safety verification, could broaden applications for both manned and unmanned aircraft. Integration of probabilistic, data-driven policies with real-time sensor feedback offers a scalable, scientifically grounded alternative to chart-based approaches, paving the way for more automated, adaptive, and safer landing operations in diverse operational and meteorological environments.

References

1. J. Fu, K. Luo, and S. Levine, “Learning robust rewards with adversarial inverse reinforcement learning,” ArXiv Prepr., 2017.
2. J. Sun, L. Yu, P. Dong, B. Lu, and B. Zhou, “Adversarial inverse reinforcement learning with self-attention dynamics model,” IEEE Robot. Autom. Lett., 2021.
3. D. Venuto, J. Chakravorty, L. Boussioux, J. Wang, G. McCracken, and D. Precup, “OIRL: Robust adversarial inverse reinforcement learning with temporally extended actions,” ArXiv Prepr., 2020.
4. S. Y. Arnob, “Off-policy adversarial inverse reinforcement learning,” ArXiv Prepr., 2020.
5. M. Zhan, J. Fan, and J. Guo, “Generative adversarial inverse reinforcement learning with deep deterministic policy gradient,” IEEE Access, 2023.
6. M. Yuan, M.-O. Pun, Y. Chen, and Q. Cao, “Hybrid adversarial inverse reinforcement learning,” ArXiv Prepr., 2021.
7. P. Wang, D. Liu, J. Chen, and C.-Y. Chan, “Adversarial inverse reinforcement learning for decision making in autonomous driving,” ArXiv Prepr., 2019.
8. P. Wang, D. Liu, J. Chen, L. Han-han, and C.-Y. Chan, “Human-like decision making for autonomous driving via adversarial inverse reinforcement learning,” ArXiv Prepr., 2019.
9. P. Wang, D. Liu, J. Chen, H. Li, and C.-Y. Chan, “Decision making for autonomous driving via augmented adversarial inverse reinforcement learning,” ArXiv Prepr., 2019.
10. L. Yu, J. Song, and S. Ermon, “Multi-agent adversarial inverse reinforcement learning,” ArXiv Prepr., 2019.
11. K. Pattanayak, V. Krishnamurthy, and C. Berry, “Inverse-inverse reinforcement learning: How to hide strategy from an adversarial inverse reinforcement learner,” in Proceedings of the IEEE Conference on Decision and Control (CDC), 2022.
12. J. Chen, T. Lan, and V. Aggarwal, “Hierarchical adversarial inverse reinforcement learning,” IEEE Trans. Neural Netw. Learn. Syst., 2023.
13. G. Xiang, S. Li, F. Shuang, F. Gao, and X. Yuan, “SC-AIRL: Share-critic in adversarial inverse reinforcement learning for long-horizon tasks,” IEEE Robot. Autom. Lett., 2024.
14. L. Zhu et al., “DRL-RNP: Deep Reinforcement Learning-Based Optimized RNP Flight Procedure Execution,” Sensors, vol. 22, no. 17, Aug. 2022, doi: 10.3390/s22176475.
15. Sun H, Zhang W, Yu R, Zhang Y. “Motion planning for mobile robots—Focusing on deep reinforcement learning: A systematic review”. IEEE Access. 2021 Apr 29;9:69061-81.
16. D. J. Richter and R. A. Calix, “QPlane: An Open-Source Reinforcement Learning Toolkit for Autonomous Fixed Wing Aircraft Simulation,” in Proceedings of the 12th ACM Multimedia Systems Conference, in MMSys ’21. New York, NY, USA: Association for Computing Machinery, Sept. 2021, pp. 261–266. doi: 10.1145/3458305.3478446.

17. W. Wang and J. Ma, "A review: Applications of machine learning and deep learning in aerospace engineering and aero-engine engineering," Adv. Eng. Innov., vol. 6, pp. 54–72, Feb. 2024, doi: 10.54254/2977-3903/6/2024060.
18. Z. Hu et al., "Deep Reinforcement Learning Approach with Multiple Experience Pools for UAV's Autonomous Motion Planning in Complex Unknown Environments," Sensors, vol. 20, no. 7, Mar. 2020, doi: 10.3390/s20071890.
19. E. C. P. Neto et al., "Deep Learning in Air Traffic Management (ATM): A Survey on Applications, Opportunities, and Open Challenges," Aerospace, vol. 10, no. 4, Apr. 2023, doi: 10.3390/aerospace10040358.
20. J. Hu, X. Yang, W. Wang, P. Wei, L. Ying, and Y. Liu, "Obstacle Avoidance for UAS in Continuous Action Space Using Deep Reinforcement Learning," IEEE Access, vol. 10, pp. 90623–90634, 2022, doi: 10.1109/ACCESS.2022.3201962.
21. H. Ali, D.-T. Pham, S. Alam, and M. Schultz, "A Deep Reinforcement Learning Approach for Airport Departure Metering Under Spatial–Temporal Airside Interactions," IEEE Trans. Intell. Transp. Syst., vol. 23, no. 12, pp. 23933–23950, Dec. 2022, doi: 10.1109/TITS.2022.3209397.
22. J. Choi et al., "Modular Reinforcement Learning for Autonomous UAV Flight Control," Drones, vol. 7, no. 7, June 2023, doi: 10.3390/drones7070418.
23. C. Chronis, G. Anagnostopoulos, E. Politi, G. Dimitrakopoulos, and I. Varlamis, "Dynamic Navigation in Unconstrained Environments Using Reinforcement Learning Algorithms," IEEE Access, vol. 11, pp. 117984–118001, 2023, doi: 10.1109/ACCESS.2023.3326435.
24. D. Wada, S. A. Araujo-Estrada, S. Windsor, D. Wada, S. A. Araujo-Estrada, and S. Windsor, "Unmanned Aerial Vehicle Pitch Control under Delay Using Deep Reinforcement Learning with Continuous Action in Wind Tunnel Test," Aerospace, vol. 8, no. 9, Sept. 2021, doi: 10.3390/aerospace8090258.
25. Azar, A.T., Koubaa, A., Ali Mohamed, N., Ibrahim, H.A., Ibrahim, Z.F., Kazim, M., Ammar, A., Benjdira, B., Khamis, A.M., Hameed, I.A. and Casalino, G., 2021. "Drone deep reinforcement learning: A review". Electronics, 10(9), p.999.
26. J. Lou et al., "Real-Time On-the-Fly Motion Planning for Urban Air Mobility via Updating Tree Data of Sampling-Based Algorithms Using Neural Network Inference," Aerospace, vol. 11, no. 1, Jan. 2024, doi: 10.3390/aerospace11010099.
27. H. Zhu et al., "Inverse Reinforcement Learning-Based Fire-Control Command Calculation of an Unmanned Autonomous Helicopter Using Swarm Intelligence Demonstration," Aerospace, vol. 10, no. 3, Mar. 2023, doi: 10.3390/aerospace10030309.
28. Brunke, L., Greeff, M., Hall, A.W., Yuan, Z., Zhou, S., Panerati, J. and Schoellig, A.P., 2022. "Safe learning in robotics: From learning-based control to safe reinforcement learning." *Annual Review of Control, Robotics, and Autonomous Systems*, *5*(1), pp.411-444.
29. E. Alipour and S. Malaek, "Learning Capable Aircraft (LCA): an Efficient Approach to Enhance Flight-Safety," under review.
30. "WMO Codes Registry : wmdr/DataFormat/FM-15-metar." Accessed: Jan. 05, 2026. [Online]. Available: https://codes.wmo.int/wmdr/DataFormat/FM-15-metar
31. "Iowa Environmental Mesonet." Accessed: Jan. 05, 2026. [Online]. Available: https://mesonet.agron.iastate.edu/cgi-bin/request/asos.py
32. "Climate trends and projections for Guam | U.S. Geological Survey." Accessed: Jan. 05, 2026. [Online]. Available: https://www.usgs.gov/publications/climate-trends-and-projections-guam
33. "Airport Design, Updates to the standards for Taxiway Fillet Design, Advisory Circular No.:150-5300-13A, FAA, 2014."
34. D. Freedman and P. Diaconis, "On the histogram as a density estimator: L2 theory," Z. Für Wahrscheinlichkeitstheorie Verwandte Geb., vol. 57, no. 4, pp. 453–476, 1981.
35. J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adversarial inverse reinforcement learning," in Proceedings of the 6th International Conference on Learning Representations (ICLR), 2018.
36. A. Y. Ng, D. Harada, and S. Russell, "Policy Invariance under Reward Transformations: Theory and Application to Reward Shaping," in Proceedings of the 16th International Conference on Machine Learning (ICML), Morgan Kaufmann, 1999, pp. 278–287.

 doi:10.20944/preprints202602.0823.v1

37. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," ArXiv Prepr. ArXiv170706347, 2017.
38. "Aircraft Characteristics Database | Federal Aviation Administration." Accessed: Dec. 31, 2025. [Online]. Available: https://www.faa.gov/airports/engineering/aircraft_char_database
39. "Use of Aircraft Approach Category During Instrument Approach Operations, FAA, 01/09/2023."