

ApproachNet: Physics-Grounded Safety Layers for Data-Driven Aircraft Approach Control

Justin Groh  and Andreas Lehrmann 

Technical University of Applied Sciences Würzburg-Schweinfurt

*Corresponding author: andreas.lehrmann@thws.de

Abstract

Automated aircraft guidance during the approach phase typically relies on optimal control or neural architectures. We combine the strengths of both paradigms with ApproachNet, a hybrid framework enforcing physical and operational hard constraints while retaining the computational efficiency of data-driven methods. Grounded in an aircraft dynamics model and control barrier functions, ApproachNet employs a neural backbone and supports end-to-end training on historical ADS-B data. We demonstrate generalization across multiple airports, runways, and aircraft types in single- and multi-aircraft approach control.

Keywords: deep learning, quadratic programming, physics-based modeling, approach control

1. Introduction

The approach remains one of the most demanding phases of flight, with high workload, dense traffic, and operations near the limits of the flight envelope contributing to a disproportionate share of incidents and accidents [1, 2]. Traditional autopilots, such as Garmin's GFC 600 [3], provide reliable guidance and control but cannot fully address these risks because they lack the operational context required for holistic approach planning. Consequently, modern flight automation increasingly relies on integrated systems modeling the entire approach trajectory [4].

Frameworks based on optimal control express the approach trajectory as a constrained objective [5, 6]. While such methods allow for an explicit specification of operational constraints, they are computationally expensive and not guaranteed to converge [7], preventing their use in time-critical in-flight scenarios. A growing body of work thus relies on data-driven modeling with neural architectures, which learn approach trajectories directly from historical ADS-B data [8, 9]. Although these methods scale more favourably to complex environments, their probabilistic nature and lack of a physical foundation can result in unsafe or infeasible trajectories. They also typically model each aircraft in isolation, ignoring the separation requirements between simultaneously approaching aircraft.

We address these challenges with ApproachNet, a deep learning architecture for approach trajectory generation that combines a modern neural backbone with explicit flight mechanics and operational hard constraints incorporated during training and enforced at inference. We first consider a single aircraft, using ApproachNet to predict low-level control parameters of an aircraft dynamics model. We demonstrate that the resulting trajectories are feasible, safe, and computable at interactive rates. We then extend ApproachNet to multiple simultaneously approaching aircraft by enabling information exchange between them and enforcing pairwise separation in a joint safety layer.

Contributions. We make four contributions: (i) a neural architecture for approach trajectory generation that integrates an aircraft dynamics model and a differentiable safety layer [10, 11] to account for physical and operational constraints; (ii) a multi-aircraft extension combining graph attention with a joint separation layer; (iii) a reusable OpenSky-based pipeline for learning approach control from ADS-B data (Appendix 2); and (iv) a comparison with state-of-the-art optimization- and learning-based approaches, demonstrating generalization across airports, runways, and aircraft types.

2. Related Work

The use of artificial intelligence in aviation includes a variety of surveillance, prediction, and optimization tasks [12]. Constrained approach trajectory planning, in particular, has been addressed using several structure-exploiting methods and nonlinear programming solvers [13, 14, 15, 16]. To reduce computational cost, fully data-driven trajectory forecasting based on ADS-B data has been extensively studied but typically lacks a physical and procedural foundation [17, 18, 19, 8]. Alternatives include reinforcement learning, which has been used for end-to-end action prediction during landing operations, albeit only for a single aircraft type and landing site [20, 21, 22]. To embed hard constraints into learned models, control barrier functions (CBFs) have been unified with performance objectives in quadratic-program (QP) formulations [10, 23]. Such CBF-QP filters have been combined with reinforcement learning [24] and applied to collision avoidance and geofencing for fixed-wing aircraft [25]. Building on differentiable optimization layers [11, 26], CBF-QPs have further been embedded as end-to-end trainable layers in neural network controllers [27, 28], while decentralized and graph-based neural barrier certificates have been learned for multi-agent settings [29, 30, 31].

3. Approach Net

ApproachNet addresses the task of guiding aircraft from the boundary of the approach region to the runway threshold by predicting explicit control commands that satisfy physical and operational constraints. Our method comprises three components: a neural backbone (ResNet-101 [32] or FT-Transformer [33]) predicting control commands; a hard constraint layer (H), formulated as a control barrier function quadratic program (CBF-QP) [10, 23], projecting these commands onto the set of safe and feasible controls; and an aircraft dynamics model (D) propagating the aircraft state forward in time ($\rightarrow$ Appendix 7.1). We first introduce the feature representation provided as input to the neural backbone, then describe the three components for a single aircraft, and finally extend the framework to multiple simultaneously approaching aircraft (Figure 1).

3.1 Model Architecture

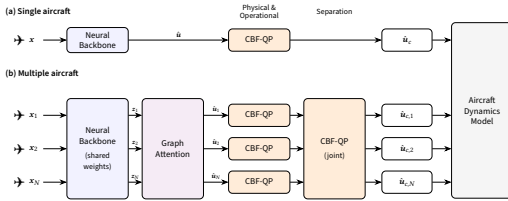


Figure 1. ApproachNet Architecture. The neural backbone is given either by a ResNet-101 [32] or a FT-Transformer [33], whereas the graph attention and QP implementations follow [34] and [26], respectively. Finally, the aircraft dynamics model ($\rightarrow$ Appendix 7.1) integrates $\hat{u}_c$ forward in time.

offsets from the airport center, with aircraft elevations expressed in meters relative to the runway threshold’s absolute orthometric height, which is obtained by subtracting the geoid undulation [36] from its WGS84 ellipsoidal height. This relative representation facilitates learning across airports. All features are subsequently z-score normalized using statistics computed from the training dataset.

Single-Aircraft Control. For a single aircraft, the neural backbone maps the feature vector $\mathbf{x}$ to predicted normalized controls $\hat{\mathbf{u}} = (T, n, \phi)^T$ comprising thrust, load factor, and bank angle. To this end, we adapt ResNet-101’s stem convolution and regression head to our input and control dimensions, and configure the FT-Transformer with an equivalent regression head on its [CLS] token. The predicted controls then pass through a hard constraint layer comprising two groups of CBF constraints: an operational group using high-order CBFs [23] to enforce lateral and vertical track-error bounds in the form of an LPV funnel (FAA Order 8260.58D [37]; RTCA DO-229D [38]), and a phys-

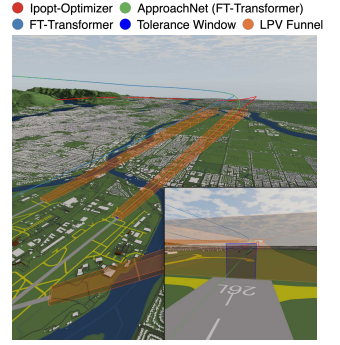
Feature Representation. The neural backbone predicts control commands based on a feature representation $\mathbf{x} \in \mathbb{R}^{15}$ used by both single- and multi-aircraft models. This representation comprises: (1) five aircraft-specific aerodynamic parameters sourced from OpenAP [35] ($\rightarrow$ Appendix 6); (2) six aircraft state variables ($\rightarrow$ Appendix 7); and (3) four runway descriptors specifying its threshold position, orthometric height, and bearing angle. Aircraft and runway positions are expressed as UTM coordinate

Table 1: Quantitative Evaluation (Single Aircraft). Approach performance by neural backbone and presence of D/H/T modules, averaged across 3 airports and 791 test approaches. succ@s = success ratio at s times reference tolerances. $AUC = \int_0^1 auc@s \, ds$. See Appendix 5 for details.

Model	D	H	T	succ@1 (%)	succ@2 (%)	succ@3 (%)	AUC (%)	Lateral Dev. (m)	Vertical Dev. (m)	HDG (°)	GP (°)	Velocity Dev. (m/s)	Landing Time (s)	Solving Time (s)
ResNet-101 [32]	-	-	-	0.25	0.38	1.39	2.72	6620.31	1331.60	93.92	32.43	73.83	1037.59	15.85
	✓	-	-	0.00	0.38	1.01	25.25	1836.37	698.25	25.66	2.43	24.06	761.74	11.70
ApproachNet (ResNet-101) (ours)	-	✓	-	2.53	4.05	4.94	7.98	5026.45	1135.53	75.67	32.19	75.82	904.99	20.38
	✓	✓	-	38.56	68.27	72.44	61.86	3823.90	161.08	15.49	2.46	62.90	511.06	10.38
	✓	✓	✓	42.66	66.94	73.29	62.43	3759.40	112.88	14.94	2.41	62.53	484.21	10.05
FT-Transformer [33]	-	-	-	0.00	0.00	0.00	0.00	5790.26	883.74	89.60	0.55	51.34	1419.11	2.79
	✓	-	-	0.00	0.38	0.63	22.96	3556.49	1127.98	40.04	2.26	32.84	1145.22	3.21
ApproachNet (FT-Transformer) (ours)	-	✓	-	0.00	0.25	0.76	12.71	2722.13	478.67	68.08	0.69	57.37	1149.93	7.72
	✓	✓	-	21.04	41.06	53.99	46.95	4998.42	352.15	26.81	3.26	86.51	533.93	4.11
	✓	✓	✓	32.87	59.04	67.76	57.30	4117.14	189.50	15.79	3.15	75.48	534.47	3.74
Ipopt-Optimizer [13]	✓	✓	-	93.68	93.68	93.68	62.70	10.42	14.56	3.56	1.53	18.73	584.83	71.67

D = aircraft dynamics model; H = hard constraint layer (CBF-QP); T = integration of H into training; Ipopt included as reference at non-interactive rates; evaluated on a single AMD Ryzen 9 7900X3D.

Figure 2: Qualitative Insights I.



ical group using a standard CBF [10] to enforce a stall-speed margin. These constraints, together with control bounds and approach information ($\rightarrow$ Appendix 3), define a differentiable CBF-QP [11] formulated in reparameterized controls with input-affine dynamics. The solution is mapped back to corrected controls $\hat{\mathbf{u}}_c$, which the aircraft dynamics model uses to integrate the equations of motion and obtain the next state. Figure 1a illustrates this pipeline.

Multi-Aircraft Control. For N simultaneously active aircraft, a shared neural backbone encodes each feature vector $\mathbf{x}_i$ into a 192-dimensional latent embedding $\mathbf{z}_i$. These embeddings interact through graph attention [34, 39] over a fully connected graph, providing each aircraft's control prediction $\hat{\mathbf{u}}_i$ with situational awareness of the surrounding traffic. As in the single-aircraft case, each predicted control first passes through its own CBF-QP. Since separation constraints couple the control inputs of multiple aircraft, a subsequent separation QP jointly refines these controls while retaining all unary constraints. Specifically, it adds one pairwise HO-CBF [23] per aircraft pair to enforce the FAA cylindrical separation minima (2.5 NM horizontal, 1000 ft vertical)¹ through a smooth superellipsoidal relaxation that fully contains the cylinder. Pairwise constraints are included only for aircraft within 10 NM of each other, since more distant pairs cannot create a conflict within the planning horizon and would unnecessarily increase the QP size [41]. The aircraft dynamics model then integrates the equations of motion using the corrected controls $\hat{\mathbf{u}}_{c,1}, \dots, \hat{\mathbf{u}}_{c,N}$ to obtain the subsequent states. See Figure 1b for an overview.

3.2 Training

We train the backbone using a loss on (T, n, ϕ) controls recovered from ADS-B trajectories via inverse dynamics ($\rightarrow$ Appendix 7.2). The loss is applied either to raw predictions $\hat{\mathbf{u}}$ (D & D+H in Tables 1 & 2) or travels through the CBF-QP modules, integrating safety layers into end-to-end training (D+H+T).

Training with Hard-Constraint Layer. Let $\mathbf{u}^{(i)} = (T^{(i)}, n^{(i)}, \phi^{(i)})$ be the control label of sample i , $\hat{\mathbf{u}}^{(i)}$ the raw prediction produced by the neural backbone, and $\Pi(\hat{\mathbf{u}}^{(i)}) = \hat{\mathbf{u}}_c^{(i)}$ its correction by the CBF-QP safety layer Π . Since the CBF-QP is differentiable [26], we can perform end-to-end training by backpropagating gradients through both the safety layer and the neural backbone. However, in situations where the constraints are highly active, Π can significantly alter the raw prediction $\hat{\mathbf{u}}^{(i)}$, which can cause training instability and convergence issues. To address this, we calculate losses on both $\hat{\mathbf{u}}^{(i)}$ and $\Pi(\hat{\mathbf{u}}^{(i)})$, with a weight reflecting the extent of intervention by the safety layer:

$$\delta^{(i)} = \frac{1}{3} \left\| \Pi(\hat{\mathbf{u}}^{(i)}) - \hat{\mathbf{u}}^{(i)} \right\|_2^2, \quad \mathbf{w}^{(i)} = m^{(i)} \cdot \exp\left(-\frac{\delta^{(i)}}{\tau}\right), \quad (1)$$

$$\mathcal{L} = \frac{1}{B} \sum_{i=1}^B \left[\mathbf{w}^{(i)} \frac{1}{3} \left\| \Pi(\hat{\mathbf{u}}^{(i)}) - \mathbf{u}^{(i)} \right\|_2^2 + \left(1 - \mathbf{w}^{(i)}\right) \frac{1}{3} \left\| \hat{\mathbf{u}}^{(i)} - \mathbf{u}^{(i)} \right\|_2^2 \right], \quad (2)$$

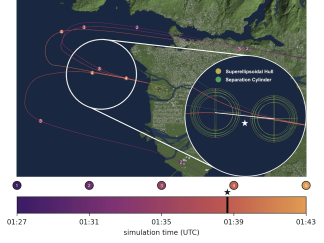
¹These thresholds serve as illustrative separation criteria [40]; context-specific requirements can be readily incorporated.

Table 2: Quantitative Evaluation (Multi-Aircraft). Approach performance by neural backbone and presence of D/H/T modules, averaged across 3 airports and 791 test approaches. Metrics as in Table 1.

Model	D	H	T	succ81 (%)	succ82 (%)	succ83 (%)	AUC (%)	Lateral Dev. (m)	Vertical Dev. (m)	HDG (°)	GP (°)	Velocity Dev. (m/s)	Landing Time (s)	Solving Time (s)
ResNet-101-GNN	✓	-	-	0.00	0.13	0.13	2.75	5537.59	3262.71	83.79	10.70	91.47	1184.18	3.82
ApproachNet (ResNet-101-GNN)	✓	✓	-	7.59	23.51	31.98	24.61	6803.53	1275.79	61.34	6.98	107.49	590.72	6.66
	✓	✓	✓	15.55	34.26	43.11	34.88	6404.16	751.30	52.90	3.74	85.62	481.70	5.87
FT-Transformer-GNN	✓	-	-	0.00	0.13	0.25	6.13	5279.85	1593.41	70.06	4.97	56.23	1066.55	2.06
ApproachNet (FT-Transformer-GNN)	✓	✓	-	13.02	26.42	34.39	28.17	6562.48	898.61	55.43	5.70	99.07	590.73	4.76
	✓	✓	✓	20.48	36.54	45.01	36.93	5549.86	633.82	42.83	3.83	90.39	549.20	4.66

D = aircraft dynamics model; H = hard constraint layer (CBF-QP + separation QP);

T = integration of H into training; evaluated on a single NVIDIA GeForce RTX 4080 Super.

Figure 3: Qualitative Insights II.

where B is the batch size, $m^{(i)} \in \{0, 1\}$ indicates if the QP for sample i was successfully solved, and $\tau > 0$ is a weighting temperature. For samples that remain unsolved, training proceeds solely based on the raw prediction. The weight $w^{(i)}$ is held constant during backpropagation by applying a stop-gradient operator. For successfully solved samples, small interventions ($\delta^{(i)} \rightarrow 0$, $w^{(i)} \rightarrow 1$) emphasize supervision through the safety layer, whereas large interventions ($w^{(i)} \rightarrow 0$) shift the focus to direct supervision of the backbone’s raw predictions. In the multi-aircraft scenario, Π denotes the composition of individual aircraft CBF-QPs and the subsequent joint separation QP, through which gradients are also propagated. The overall loss is computed as an average over all aircraft and time steps within the batch after a warm-start training of the backbone with the raw-prediction loss.

Choice of the Temperature. The temperature τ controls how quickly the weight $w^{(i)}$ decays with the size of the intervention. We calibrate τ once before training: each ground truth label is passed through the safety layer, and the resulting normalized label shift $\Delta^{(i)}$ measures how strongly the constraints disagree with the demonstrated behaviour. We then choose τ such that a label shift at the q -quantile $\Delta_{(q)}$ of $\{\Delta^{(i)}\}_i$ receives a prescribed reference weight w_{ref} . Note that w_{ref} is a calibration constant only; during training, τ is kept fixed and the sample-specific weights $w^{(i)}$ follow Equation 1.

$$\Delta^{(i)} = \frac{1}{3} \left\| \Pi(\mathbf{u}^{(i)}) - \mathbf{u}^{(i)} \right\|_2^2, \quad \exp\left(-\frac{\Delta_{(q)}}{\tau}\right) = w_{\text{ref}} \implies \tau = \frac{\Delta_{(q)}}{\ln(1/w_{\text{ref}})}, \quad q = 0.9, \quad w_{\text{ref}} = 0.15. \quad (3)$$

4. Experiments

We evaluate ApproachNet’s single- and multi-aircraft models on a test dataset comprising 791 approach trajectories across 3 airport geometries and 36 aircraft types, with trajectories starting approximately 30 km from the respective airport; see Appendix 2 for data details. We compare against two data-driven baselines [32, 33] and perform ablation studies assessing the contributions of flight mechanics (D), inference with hard constraints (H), and training with hard constraints (T). The results demonstrate favourable performance in both the single-aircraft (Table 1) and multi-aircraft (Table 2) settings; see Appendix 5 for the evaluation tolerances. For reference, we also include Ipopt as an optimal control baseline [13, 14], which achieves a higher approach success rate but requires substantially more computation time and does not support the dynamic addition of aircraft, limiting its suitability for time-critical applications compared with ApproachNet’s interactive inference speed. We supplement these quantitative results with visualizations of approach trajectories illustrating adherence to LPV bounds (Figure 2) and separation criteria (Figure 3).

5. Conclusion

We showed that enforcing hard constraints via a differentiable CBF-QP safety layer grounded in an aircraft dynamics model consistently outperforms purely data-driven neural architectures for approach control while keeping the predicted trajectories feasible, safe, and efficient. Extending the safety layer with a separation CBF-QP and the backbone with graph attention transfers these benefits to multiple simultaneously approaching aircraft, with the integration of the safety layer into training being particularly beneficial. Future work could investigate synthetically generated approaches to enhance generalisation, as well as further assess scalability in increasingly dense approach scenarios.

Appendix 1. Author Contributions

- Justin Groh: conceptualization, data collection, implementation, evaluation, and writing/editing.
- Andreas Lehrmann: conceptualization, technical foundation, and writing/editing.

Appendix 2. Open Data Statement

We use ADS-B trajectories from the OpenSky Network, accessed through its Trino cluster [42]; digital elevation data from the Copernicus Data Space Ecosystem²; runway geometry from the X-Plane Scenery Gateway³; additional airport infrastructure, including taxiways and terminal buildings, from OpenStreetMap⁴; and approach plates and fixes from the AIRAC API⁵. Aircraft types are identified from ICAO 24-bit addresses using the adsbdb API⁶, and their performance parameters are obtained from OpenAP [35]. All data is publicly available⁷ and has been anonymized by replacing ICAO 24-bit addresses, Mode-S codes, and call signs with keyed HMAC-SHA256 hashes, and by removing any auxiliary identifiers (registration, owner, squawk, receiver serials, photo URLs).

Contributions to the OpenSky community. This work contributes four reusable resources for the OpenSky community: (i) an *approach extraction pipeline* that queries the OpenSky Trino database and associates each trajectory with its runway and approach procedure (→ Appendix 3); (ii) *ADS-B-derived control labels* obtained through an inverse dynamics model (→ Appendix 7.2); (iii) a *reproducible benchmarking setup* with fixed train/test splits, evaluation metrics, and tolerances (→ Appendix 5); and (iv) *data preprocessing and filtering routines* integrated into the pipeline to address OpenSky's low-altitude coverage gaps, correct geometric altitudes using geoid undulation, and mitigate noise in derived rates.

Appendix 3. Dataset and Preprocessing

The dataset covers ADS-B trajectories within a radius of 30 km around each airport. Across the three evaluated airports, the training set contains 12 252 approaches, comprising 7 360 173 aircraft states. Trajectories were selected and processed as follows:

1. Only flights in the approach phase were considered.
2. Only complete trajectories that reached a runway were retained.
3. Irregularly sampled trajectories were resampled at a fixed rate of 1 Hz.

Each trajectory also includes auxiliary information passed to the CBF-QP safety layer:

1. The assigned landing runway and approach procedure, from which the intermediate fix (IF) and final approach fix (FAF) positions, runway threshold position, runway bearing, threshold crossing height (TCH), and runway length are extracted.
2. Aircraft-type-specific aerodynamic parameters (→ Appendix 6).
3. Aircraft-type-specific control limits ($T_{\max}$, $n_{\min}$, $n_{\max}$, $\phi_{\max}$).

²<https://dataspace.copernicus.eu/>

³<https://gateway.x-plane.com>

⁴<https://www.openstreetmap.org>

⁵<https://airac.net/>

⁶<https://www.adsbdb.com>

⁷https://projects.cairo.thwms.de/asml/papers/approach_net/

Appendix 4. Reproducibility Statement

A repository with our code and scripts reproducing our quantitative results is publicly available.⁸

Appendix 5. Reference Tolerances

succ@s and auc@s measure the fraction of successful approaches at scale s . We consider an approach at scale s successful if all five component deviations are less than s times the reference tolerances in Table 3 used to calculate a successful approach at scale 1. All deviations are measured at the aircraft's closest point to the runway threshold.

Table 3. Reference Tolerances. We show the reference tolerances used to calculate succ@1 and auc@1; tolerances for other values of s can be obtained by linear scaling.

Metric	Reference	Tolerance	
		succ@1	auc@1
Lateral deviation	centerline	15 m	2500 m
Vertical deviation	threshold crossing height	20 m	1000 m
Heading (HDG; χ) deviation	runway bearing	10°	10°
Glide path (GP; γ) deviation	-3°	3°	3°
Approach speed deviation	1.3 V_{stall}	50 %	250 %

Appendix 6. Aircraft Specification

Each aircraft type is characterized by the following parameters, sourced and calculated from OpenAP [35] and used in both the backbone's feature representation and the aircraft dynamics model (→ Appendix 7):

Table 4. Aircraft Specification. Definitions of aircraft mass and aerodynamic parameters.

Variable	Unit	Description	Included in Feature Representation
m	kg	Aircraft landing mass	✓
S	m ²	Wing planform area	✓
C_L	dimensionless	Lift coefficient	✗
$C_{L,\text{max}}$	dimensionless	Maximum lift coefficient	✓
C_D	dimensionless	Drag coefficient	✗
C_{D_0}	dimensionless	Zero-lift drag coefficient	✓
k	dimensionless	Induced drag factor in $C_D = C_{D_0} + k C_L^2$	✓

Aerodynamic parameters refer to the landing configuration.

⁸https://projects.cairo.thws.de/asml/papers/approach_net/

Appendix 7. Aircraft Dynamics Model

Appendix 7.1 Forward Dynamics Model

We model each aircraft using a translational point-mass model under zero-wind conditions. The state $\mathbf{s} = [x, y, h, v, \gamma, \chi]^T$ comprises the position (x, y, h) , the true airspeed v , the flight path angle γ , and the track angle χ . The control $\mathbf{u} = [T, n, \phi]^T$ comprises thrust T , load factor $n = L/(mg)$, and bank angle ϕ . The coordinates x , y , and h point east, north, and upward, respectively, and χ is measured clockwise from north.

Decomposing the velocity along γ and χ gives

$$\mathbf{v} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{h} \end{bmatrix} = v \begin{bmatrix} \cos \gamma \sin \chi \\ \cos \gamma \cos \chi \\ \sin \gamma \end{bmatrix}, \quad (4)$$

with unit flight direction $\mathbf{t} = \mathbf{v}/\|\mathbf{v}\|$. Assuming thrust and drag act along the flight direction, Newton's second law yields

$$\dot{\mathbf{v}} = \frac{T-D}{m} \mathbf{t} + \frac{L}{m} \mathbf{n}_\phi + \mathbf{g}, \quad \mathbf{n}_\phi = \mathbf{n}_0 \cos \phi + (\mathbf{t} \times \mathbf{n}_0) \sin \phi, \quad (5)$$

with

$$\mathbf{g} = [0, 0, -g]^T, \quad \mathbf{n}_0 = \mathbf{b} \times \mathbf{t}, \quad \mathbf{b} = \frac{\mathbf{t} \times \hat{\mathbf{z}}}{\|\mathbf{t} \times \hat{\mathbf{z}}\|}, \quad \hat{\mathbf{z}} = [0, 0, 1]^T. \quad (6)$$

The lift direction $\mathbf{n}_\phi$ is thus obtained by rotating the zero-bank lift direction $\mathbf{n}_0$ about $\mathbf{t}$ through the bank angle ϕ . Drag follows $D = \frac{1}{2} \rho(h) v^2 S (C_{D_0} + k C_L^2)$, with $\rho(h) = \rho_0 e^{-h/H}$, where ρ_0 is the air density at sea level, h is the altitude above sea level, and H is the atmospheric density scale height. The lift coefficient is $C_L = 2nm g/(\rho(h) v^2 S)$, with $0 \leq n \leq \rho(h) v^2 S C_{L,\max}/(2mg)$ enforcing nonnegative lift and the stall limit.

Differentiating $\chi = \text{atan2}(v_x, v_y)$ and $\gamma = \arcsin(v_z/v)$ gives

$$\dot{\chi} = \frac{v_y \dot{v}_x - v_x \dot{v}_y}{v_h^2}, \quad \dot{\gamma} = \frac{v \dot{v}_z - v_z \dot{v}}{v^2 \cos \gamma}, \quad v_h = v \cos \gamma, \quad (7)$$

where $\dot{\mathbf{v}} = \mathbf{t}^T \dot{\mathbf{v}}$.

Appendix 7.2 Inverse Dynamics Model

To obtain supervision for the neural backbone, we estimate model-consistent control labels (T, n, ϕ) from ADS-B trajectories. We first estimate $\dot{\mathbf{v}}$, $\dot{\gamma}$, and $\dot{\chi}$ from smoothed trajectories. We then recover the bank angle ϕ from the angular rates and invert Equation 5 by projecting the non-gravitational acceleration $\dot{\mathbf{v}} - \mathbf{g}$ onto the lift and flight directions. Since the lift direction $\mathbf{n}_\phi$ depends on ϕ and drag D depends on n , the inverse model is evaluated sequentially in the order $\phi \rightarrow n \rightarrow T$, with D evaluated using the recovered load factor.

$$\phi = \text{atan2}(v_h \dot{\chi}, v \dot{\gamma} + g \cos \gamma), \quad (8)$$

$$n = \frac{(\dot{\mathbf{v}} - \mathbf{g}) \cdot \mathbf{n}_\phi}{g}, \quad (9)$$

$$T = m((\dot{\mathbf{v}} - \mathbf{g}) \cdot \mathbf{t}) + D. \quad (10)$$

Appendix 8. Additional Qualitative Insights

213

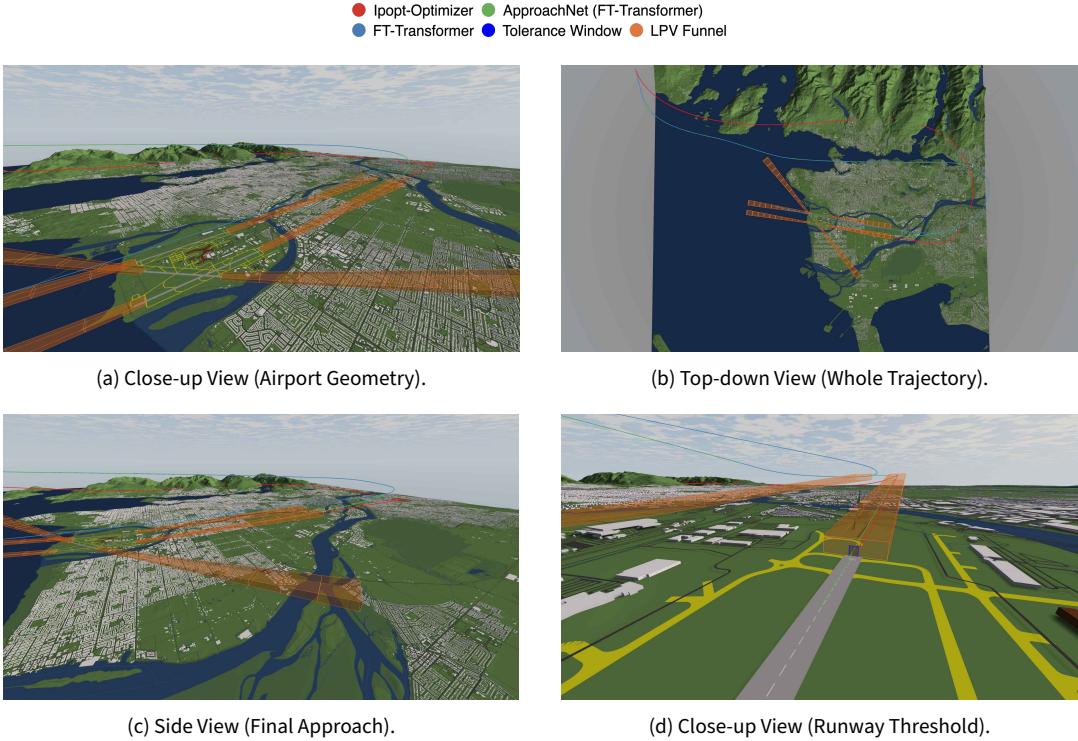


Figure 4. We show auto-regressively generated trajectories of a single approach from multiple camera views.

References

214

- [1] National Transportation Safety Board. *NTSB Investigation Update: United Flight 169, Boeing 767-400, Newark Liberty International Airport*. Tech. rep. National Transportation Safety Board, 2026. 215
- [2] Boeing Commercial Airplanes. *Statistical Summary of Commercial Jet Airplane Accidents: World-wide Operations*. Tech. rep. Boeing Commercial Airplanes, 2019. 216
- [3] Garmin International Inc. *GFC 600 Pilot's Guide*. Garmin International, Inc. 2021. 217
- [4] Alessandro Gardi, Roberto Sabatini, and Trevor Kistan. "Multiobjective 4D Trajectory Optimization for Integrated Avionics and Air Traffic Management Systems". In: *IEEE Transactions on Aerospace and Electronic Systems* (2019). 218
- [5] Taewan Kim, Abhinav Kamath, Niyousha Rahimi, Jasper Corleis, Behçet Açıkmeşe, and Mehran Mesbahi. "Six-Degree-of-Freedom Aircraft Landing Trajectory Planning with Runway Alignment". In: *Journal of Guidance, Control, and Dynamics* (2025). 219
- [6] Yutian Pang, Daniel Delahaye, and John-Paul Clarke. "Trajectory-Based Optimization for Air Traffic Control in the Terminal Maneuvering Area". In: *Transportation Research Part C: Emerging Technologies* (2027). 220
- [7] John T. Betts. "Survey of numerical methods for trajectory optimization". In: *Journal of Guidance, Control, and Dynamics* (1998). 221
- [8] Timoth   Krauth, Adrien Lafage, J  r  me Morio, Xavier Olive, and Manuel Walt  rt. "Deep generative modelling of aircraft trajectories in terminal maneuvering areas". In: *Machine Learning with Applications* (2023). 222

223

- [9] Jun Xiang and Jun Chen. "Data-driven Probabilistic Trajectory Learning with High Temporal Resolution in Terminal Airspace". In: *Journal of Aerospace Information Systems* (2024). 235
- [10] Aaron D. Ames, Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. "Control Barrier Function Based Quadratic Programs for Safety Critical Systems". In: *IEEE Transactions on Automatic Control* (2017). 236
- [11] Brandon Amos and J. Zico Kolter. "OptNet: Differentiable Optimization as a Layer in Neural Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. 2017. 237
- [12] Zhuoming Du, Jiaxuan Wu, Yuanfei Leng, and Sebastian Wandelt. "AI4ATM: A review on how Artificial Intelligence paves the way towards autonomous Air Traffic Management". In: *Journal of the Air Transport Research Society* (2025). 238
- [13] Andreas Wächter and Lorenz Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical Programming* (2005). 239
- [14] Joel Andersson, Joris Gillis, Greg Horn, James Rawlings, and Moritz Diehl. "CasADi: a software framework for nonlinear optimization and optimal control". In: *Mathematical Programming Computation* (2018). 240
- [15] Lander Vanroye, Ajay Sathya, Joris De Schutter, and Wilm Decré. "Fatrop: A fast constrained optimal control problem solver for robot trajectory optimization and control". In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2023. 241
- [16] Junzi Sun. "OpenAP.top: Open Flight Trajectory Optimization for Air Transport and Sustainability Research". In: *Aerospace* (2022). 242
- [17] Alevizos Bastas, Theocharis Kravaris, and George Vourros. "Data Driven Aircraft Trajectory Prediction with Deep Imitation Learning". In: *arXiv preprint arXiv:2005.07960* (2020). 243
- [18] Hesam Shafienya and Amelia C. Regan. "4D flight trajectory prediction using a hybrid Deep Learning prediction method based on ADS-B technology: a case study of Hartsfield-Jackson Atlanta International Airport (ATL)". In: *Transportation Research Part C: Emerging Technologies* (2022). 244
- [19] Kaiwei Luo and Jiliu Zhou. "Large Language Models for Single-Step and Multi-Step Flight Trajectory Prediction". In: *arXiv preprint arXiv:2501.17459* (2025). 245
- [20] Mohsen Zahmatkesh, Seyyed Ali Emami, Afshin Banazadeh, and Paolo Castaldi. "Robust Auto-landing Control of an agile Regional Jet Using Fuzzy Q-learning". In: *arXiv preprint arXiv:2302.10997* (2023). 246
- [21] Chi Tang and Ying-Chih Lai. "Deep Reinforcement Learning Automatic Landing Control of Fixed-Wing Aircraft Using Deep Deterministic Policy Gradient". In: *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2020. 247
- [22] Zhuang Wang, Hui Li, Haolin Wu, Feng Shen, and Ruixuan Lu. "Design of Agent Training Environment for Aircraft Landing Guidance Based on Deep Reinforcement Learning". In: *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*. 2018. 248
- [23] Wei Xiao and Calin Belta. "Control barrier functions for systems with high relative degree". In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. 2019. 249
- [24] Richard Cheng, Gábor Orosz, Richard Murray, and Joel Burdick. "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2019. 250
- [25] Tamas G. Molnar, Suresh K. Kannan, James Cunningham, Kyle Dunlap, Kerianne L. Hobbs, and Aaron D. Ames. "Collision Avoidance and Geofencing for Fixed-Wing Aircraft With Control Barrier Functions". In: *IEEE Transactions on Control Systems Technology* (2025). 251
- [26] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and Z. Kolter. "Differentiable Convex Optimization Layers". In: *Advances in Neural Information Processing Systems*. 2019. 252
- [27] Wei Xiao, Tsun-Hsuan Wang, Ramin Hasani, Makram Chahine, Alexander Amini, Xiao Li, and Daniela Rus. "BarrierNet: Differentiable Control Barrier Functions for Learning of Safe Robot Control". In: *IEEE Transactions on Robotics* (2023). 253

- [28] Hengbo Ma, Bike Zhang, Masayoshi Tomizuka, and Koushil Sreenath. “Learning Differentiable Safety-Critical Control using Control Barrier Functions for Generalization to Novel Environments”. In: *2022 European Control Conference (ECC)*. 2022. 285–286
- [29] Zengyi Qin, Kaiqing Zhang, Yuxiao Chen, Jingkai Chen, and Chuchu Fan. “Learning Safe Multi-Agent Control with Decentralized Neural Barrier Certificates”. In: *International Conference on Learning Representations*. 2021. 287–289
- [30] Songyuan Zhang, Kunal Garg, and Chuchu Fan. “Neural Graph Control Barrier Functions Guided Distributed Collision-avoidance Multi-agent Control”. In: *7th Annual Conference on Robot Learning*. 2023. 290–293
- [31] Songyuan Zhang, Oswin So, Kunal Garg, and Chuchu Fan. “GCBF+: A Neural Graph Control Barrier Function Framework for Distributed Safe Multiagent Control”. In: *IEEE Transactions on Robotics* (2025). 294–296
- [32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. 297–299
- [33] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. “Revisiting Deep Learning Models for Tabular Data”. In: *NeurIPS*. 2021. 300–301
- [34] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. “Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification”. In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. 2021. 302–304
- [35] Junzi Sun, Jacco M. Hoekstra, and Joost Ellerbroek. “OpenAP: An open-source aircraft performance model for air transportation studies and simulations”. In: *Aerospace* (2020). 305–306
- [36] Nikolaos K. Pavlis, Simon A. Holmes, Steve C. Kenyon, and John K. Factor. “The development and evaluation of the Earth Gravitational Model 2008 (EGM2008)”. In: *Journal of Geophysical Research: Solid Earth* (2012). 307–309
- [37] Federal Aviation Administration. *Order 8260.58D: United States Standard for Performance Based Navigation (PBN) Instrument Procedure Design*. Tech. rep. Federal Aviation Administration, 2025. 310–312
- [38] RTCA, Inc. *DO-229D: Minimum Operational Performance Standards for Global Positioning System/Wide Area Augmentation System Airborne Equipment*. Tech. rep. RTCA, Inc., 2006. 313–314
- [39] Matthias Fey and Jan E. Lenssen. “Fast Graph Representation Learning with PyTorch Geometric”. In: *ICLR Workshop on Representation Learning on Graphs and Manifolds*. 2019. 315–316
- [40] Federal Aviation Administration. *Order JO 7110.65BB: Air Traffic Control*. Tech. rep. Federal Aviation Administration, 2026. 317–318
- [41] Li Wang, Aaron D. Ames, and Magnus Egerstedt. “Safety Barrier Certificates for Collisions-Free Multirobot Systems”. In: *IEEE Transactions on Robotics* (2017). 319–320
- [42] Matthias Schäfer, Martin Strohmeier, Vincent Lenders, Ivan Martinovic, and Matthias Wilhelm. “Bringing up OpenSky: A large-scale ADS-B sensor network for research”. In: *IPSN-14 proceedings of the 13th international symposium on information processing in sensor networks*. 2014. 321–323