SUPPORTING EUROPEAN AVIATION

EUROCONTROL

A Neural ODE Approach to Aircraft Flight Dynamics Modelling



SESAR Innovation Days 2025

03-12-2025

Gabriel JARRY, Ramon DALMAU,
Xavier OLIVE, Philippe VERY

ONERA THE FRENCH AEROSPACE LAB

CONSORTIUM COORDINATOR sesar DEPLOYMENT MANAGER

FOUNDING MEMBER sesar JOINT UNDERTAKING

NETWORK MANAGER

Why Generate Realistic Flight Trajectories?

🌍 **Environmental impact assessment:** Estimate fuel burn, emissions, and noise under realistic operations.

📉 **Operational optimization:** Support eco-efficient trajectories, fuel-saving procedures, and climate-optimized routing.

📡 **Air traffic management research:** Simulate and evaluate new procedures or airspace designs, etc…

Modelling Approaches for trajectory generation and simulation

Physics based

BADA, OpenAP, Poll-Schumann, …

- ✅ Physically interpretable and robust
- ⚠️ But the real world is often more complex.

Hybrid



consistency !

Data based

VAE, GANs, Seq2seq (research)

- ✅ Reproduces actual distributions and patterns
- ⚠️ May violate physical laws or produce unrealistic results

Open Source, Transparency, and Reproducibility



Close Proprietary

Manufacturers models

🔒 **Access restricted** — data and models not shareable

⚙️ **Opaque methods** — cannot inspect or reproduce results

Licence based

BADA, …

🧩 **Licensed access** — available under agreement, training pipeline not available

⚙️ **Transparent physics, hidden** — equations open but calibration data proprietary

Open Models

Poll-Schumann, …

🧩 **Public results, closed process** — outputs shared but training pipeline not available

⚙️ **Limited reproducibility** — users must recode the training pipeline from papers and access similar data

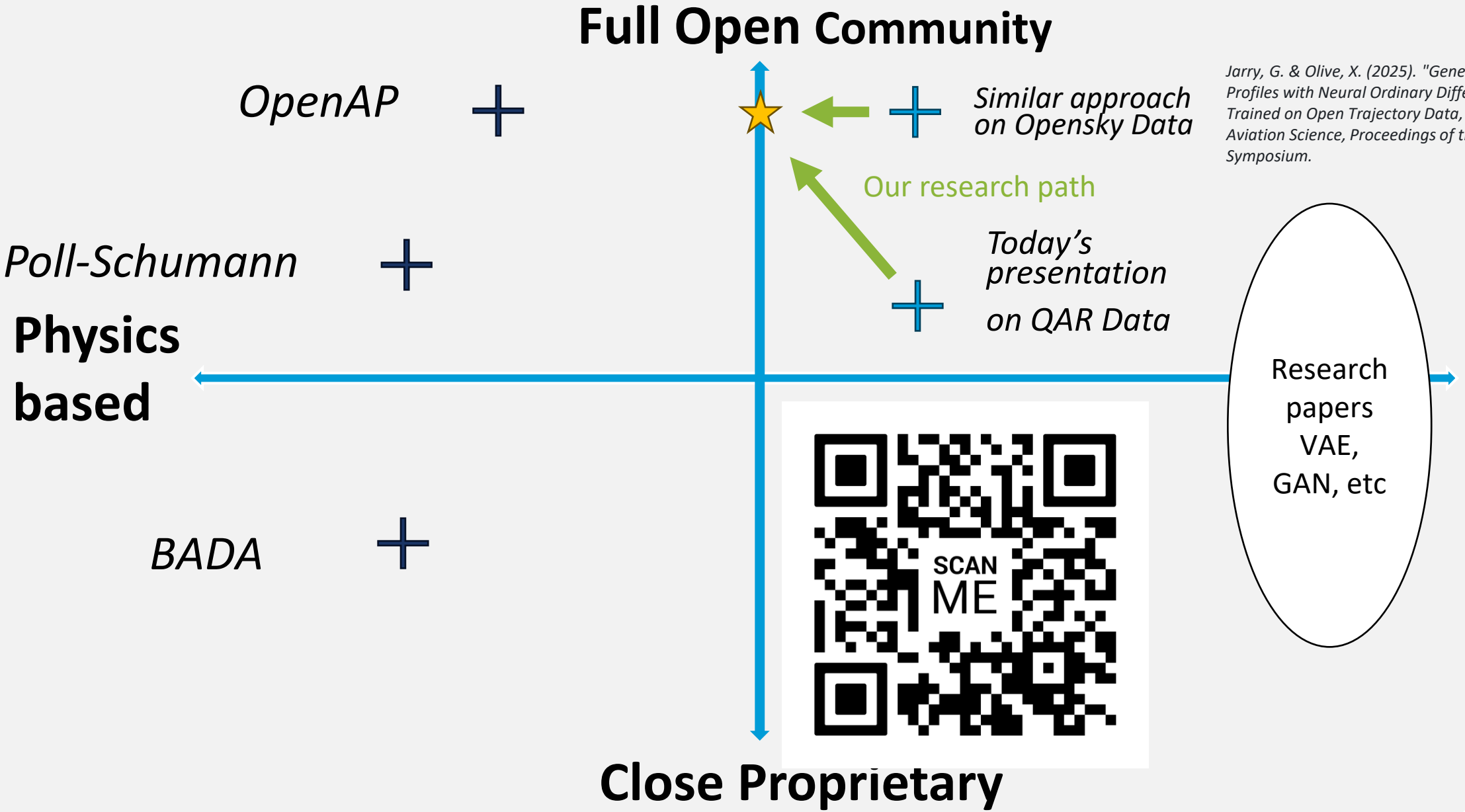
Full Open Community

OpenAP, …

🧩 **Fully transparent workflow** — model, code, and data openly available

⚙️ **High reproducibility** — anyone can retrain, verify, or extend the model

Maps of existing solutions



Jarry, G. & Olive, X. (2025). "Generation of Vertical Profiles with Neural Ordinary Differential Equations Trained on Open Trajectory Data," Journal of Open Aviation Science, Proceedings of the 13th OpenSky Symposium.

Research problems

Can Neural ODEs be trained to learn accurate and physically consistent aircraft dynamics using QAR data?

1) Neural Ordinary Differential Equations
2) Data & Model
3) Baseline & Results
4) Limitations & Conclusions

Outlines

Can Neural ODEs be trained to learn accurate and physically consistent aircraft dynamics using QAR data?

1) Neural Ordinary Differential Equations
2) Data & Model
3) Baseline & Results
4) Limitations & Conclusions

From Flight Dynamics …

The aircraft motion follows analytical equations of motion, where f represents the deterministic flight dynamics derived from physics:

$$\frac{d\boldsymbol{x}}{dt} = f(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{e}(t))$$

$\boldsymbol{x}$: state (altitude, speed, flight path angle…)
$\boldsymbol{u}$: control (altitude selected, speed selected etc…)
$\boldsymbol{e}$: environment (wind, temperature…)

… to neural ODE (simple no?!)

The aircraft motion follows analytical equations of motion, where f represents the deterministic flight dynamics derived from physics:

$$\frac{d\boldsymbol{x}}{dt} = f_{\theta}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{e}(t)\right)$$

$\boldsymbol{x}$: state (altitude, speed, flight path angle…)
$\boldsymbol{u}$: control (altitude selected, speed selected etc…)
$\boldsymbol{e}$: environment (wind, temperature…)

Neural ODE advantages

✈️ **Lagrangian perspective:** follows each aircraft *along its own path* through time.

⚙️ **Continuous and differentiable:** matches the physics of flight motion.

🧩 **Hybrid flexibility:** can combine analytical kinematics + learned residuals.

Outlines

Can Neural ODEs be trained to learn accurate and physically consistent aircraft dynamics with only open surveillance data (ADS-B and extended Mode S) ?

1) Neural Ordinary Differential Equations
2) Data & Model
3) Baseline & Results
4) Limitations & Conclusions

Dataset – A320 – QAR data



Dataset	Train	Val	Test
# Flights	5000	500	500
# Unique Airframe	5	2	2

TABLE I: List of parameters used in the model with original QAR units and converted SI units.

Feature	Symbol	QAR Unit	SI Unit	Description
State variables $x(t)$				
altitude	h	ft	m	Standard altitude above mean sea level
distance along track	d	nm	m	Along-track distance from departure
flight path angle	γ	deg	rad	Angle between velocity vector and horizon
true airspeed	V_{TAS}	kt	m/s	Speed relative to the surrounding air mass
mass	m	kg	kg	Estimated aircraft mass
Control variables $u(t)$				
selected altitude	h_{sel}	ft	m	Altitude target set in the FMS/autopilot
selected speed	V_{sel}	kt	m/s	Speed target set in the FMS/autopilot
selected vertical speed	$V_{z,\mathsf{sel}}$	ft/min	m/s	Vertical speed target set in the FMS/autopilot (if set else 0.0)
flap setting	f	index	index	Current flap configuration 0,1,2,3,4 (FULL)
landing gear status	g	binary	binary	Gear up (0) / down (1)
speed brake command	SB	binary	binary	Speed brake retracted (0) / deployed (1)
Context and intermediate variables $e(t)$				
Context variables $e_0(t)$				
outside air temperature	T_{OAT}	°C	K	Ambient static air temperature
headwind component	$V_{\parallel}$	kt	m/s	Wind component along the trajectory
crosswind component	$V_{\perp}$	kt	m/s	Wind component orthogonal to the trajectory
Trajectory variables $e_1(t)$				
Mach number	M	–	–	Ratio of V_{TAS} to speed of sound
calibrated airspeed	V_{CAS}	kt	m/s	Airspeed derived from dynamic pressure
vertical speed	V_z	ft/min	m/s	Rate of climb or descent
ground speed	V_{GS}	kt	m/s	Speed relative to the ground
selected altitude difference	$h_{\mathsf{sel}} - h$	ft	m	Difference between selected and current altitude
selected speed difference	$V_{\mathsf{sel}} - V_{\mathsf{CAS}}$	kt	m/s	Difference between selected speed and CAS
Aircraft angles $e_2(t)$				
angle of attack	α	deg	rad	Angle between chord line and relative wind
pitch angle	θ	deg	rad	Angle between aircraft longitudinal axis and horizon
Engine parameters $e_3(t)$				
N1 (fan speed)	N_1	%RPM	%RPM	Low-pressure compressor rotation speed
fuel flow	m_{fuel}	kg/h	kg/s	Total fuel consumption rate

Model architecture and formulations

$$V_z = V_{\mathrm{TAS}} \cdot \sin(\gamma),$$

$$x(t) = \{h(t), d(t), \gamma(t), V_{\mathrm{TAS}}(t), m(t)\}, \quad M = \frac{V_{\mathrm{TAS}}}{a}, \qquad a = \sqrt{\gamma_{air}\, R\, T},$$

$$V_{\mathrm{GS}} = V_{\mathrm{TAS}} - V_{\parallel},$$

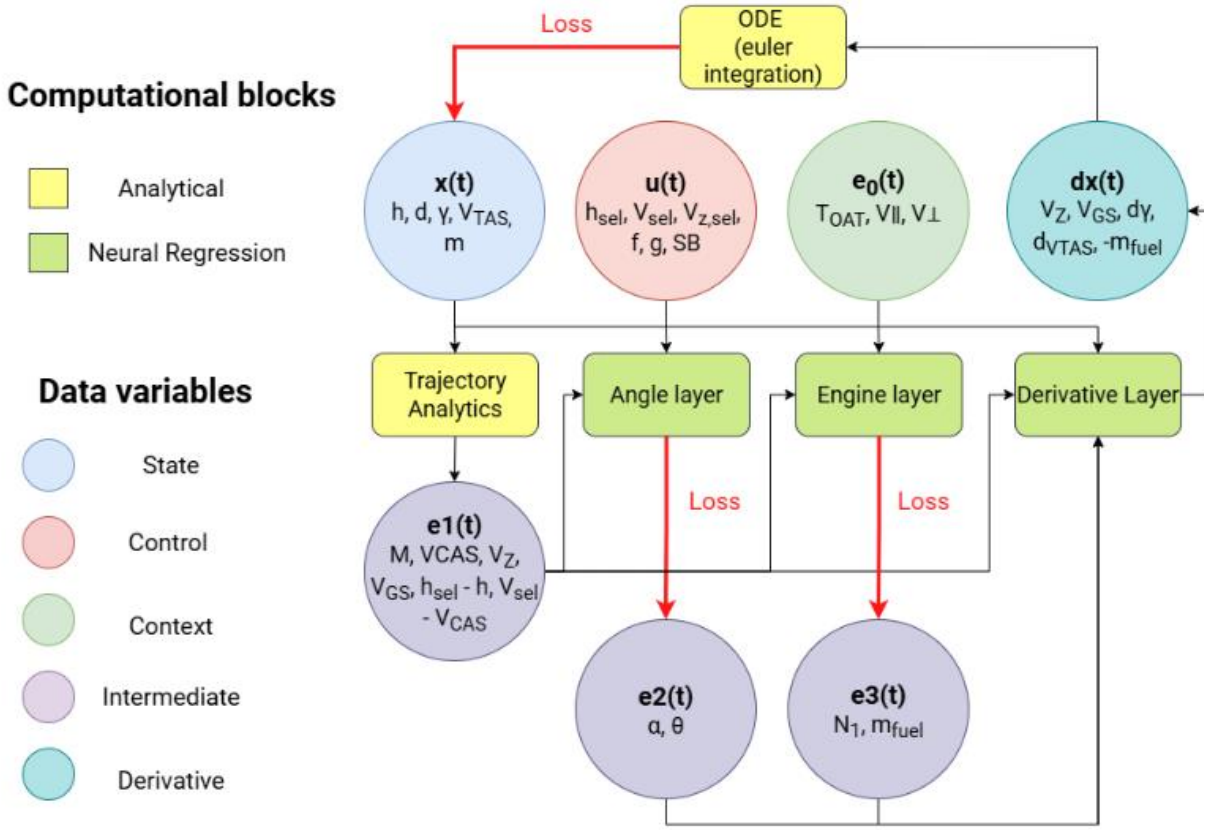


Figure 1: Architecture of the proposed model

TABLE I: List of parameters used in the model with original QAR units and converted SI units.

Feature	Symbol	QAR Unit	SI Unit	Description
State variables $x(t)$				
altitude	h	ft	m	Standard altitude above mean sea level
distance along track	d	nm	m	Along-track distance from departure
flight path angle	γ	deg	rad	Angle between velocity vector and horizon
true airspeed	V_{TAS}	kt	m/s	Speed relative to the surrounding air mass
mass	m	kg	kg	Estimated aircraft mass
Control variables $u(t)$				
selected altitude	h_{sel}	ft	m	Altitude target set in the FMS/autopilot
selected speed	V_{sel}	kt	m/s	Speed target set in the FMS/autopilot
selected vertical speed	$V_{z,\mathrm{sel}}$	ft/min	m/s	Vertical speed target set in the FMS/autopilot (if set else 0.0)
flap setting	f	index	index	Current flap configuration 0,1,2,3,4 (FULL)
landing gear status	g	binary	binary	Gear up (0) / down (1)
speed brake command	SB	binary	binary	Speed brake retracted (0) / deployed (1)
Context and intermediate variables $e(t)$				
Context variables $e_0(t)$				
outside air temperature	T_{OAT}	°C	K	Ambient static air temperature
headwind component	$V_{\parallel}$	kt	m/s	Wind component along the trajectory
crosswind component	$V_{\perp}$	kt	m/s	Wind component orthogonal to the trajectory
Trajectory variables $e_1(t)$				
Mach number	M	–	–	Ratio of V_{TAS} to speed of sound
calibrated airspeed	V_{CAS}	kt	m/s	Airspeed derived from dynamic pressure
vertical speed	V_z	ft/min	m/s	Rate of climb or descent
ground speed	V_{GS}	kt	m/s	Speed relative to the ground
selected altitude difference	$h_{\mathrm{sel}} - h$	ft	m	Difference between selected and current altitude
selected speed difference	$V_{\mathrm{sel}} - V_{\mathrm{CAS}}$	kt	m/s	Difference between selected speed and CAS
Aircraft angles $e_2(t)$				
angle of attack	α	deg	rad	Angle between chord line and relative wind
pitch angle	θ	deg	rad	Angle between aircraft longitudinal axis and horizon
Engine parameters $e_3(t)$				
N1 (fan speed)	N_1	%RPM	%RPM	Low-pressure compressor rotation speed
fuel flow	m_{fuel}	kg/h	kg/s	Total fuel consumption rate

Structured layer architecture

- Input normalization: All inputs (state, control, and context) are normalized and concatenated into a unified latent representation.
- Shared backbone: A 2-layer fully connected network (24 neurons per layer, ReLU activations) extracts common dynamical features.
- Output heads: one fully connected layers (24 neurons) with regression head
- Denormalization layer: Outputs are rescaled back to physical units to preserve interpretability and numerical stability.
- 🚫 **No dropout**

Data preparation and training process

- **Flight preparation:** subsample uniformly every 4s, convert to SI units.
- **Segment splitting and filtering:** Split each flight into non-overlapping 4-minute (60-point) segments and discard those with inconsistent or missing values.
- **Integration:** Each batch (512 samples) is integrated over 60 time steps using control inputs and the *torchdiffeq* (with Euler ODE solver).
- **Loss:** Composite loss computed as the sum of normalized MSE across all state variables within the batch ($\frac{1}{\sigma}$ Kendall & Gal, 2018).

$$\mathcal{L} = \sum_{i \in C} \alpha_i \, \ell_i\left(y_i^{\text{pred}}, y_i^{\text{true}}\right)$$

- **Backpropagation**: Adjoint gradients computed by torchdiffeq; parameters updated using AdamW.

Outlines

Can Neural ODEs be trained to learn accurate and physically consistent aircraft dynamics using QAR data?

1) Neural Ordinary Differential Equations
2) Data & Model
3) Baseline & Results
4) Limitations & Conclusions

Bada baseline estimation using pyBADA 4.2

🛬 **Initialization:**
Load aircraft-specific BADA 4.2 parameters (thrust, drag, mass assumptions) using the pyBADA interface.

⚙️ **Input mapping:**
From QAR data, extract current state and control targets.

🔁 **Routine selection logic:**
For each 4 s step, choose the appropriate TCL.py routine based on the flight phase and control targets:

- constantSpeedLevel() → cruise or steady flight
- constantSpeedROCD_time() → climb/descent with vertical-speed target
- constantSpeedRating_time() → thrust-limited segments
- accDec_time() → acceleration or deceleration phases

📈 **Trajectory reconstruction:**
Integrate each 4 s step and concatenate results (altitude, speed, vertical rate, mass) to generate a **physics-based reference trajectory**.

Overall results

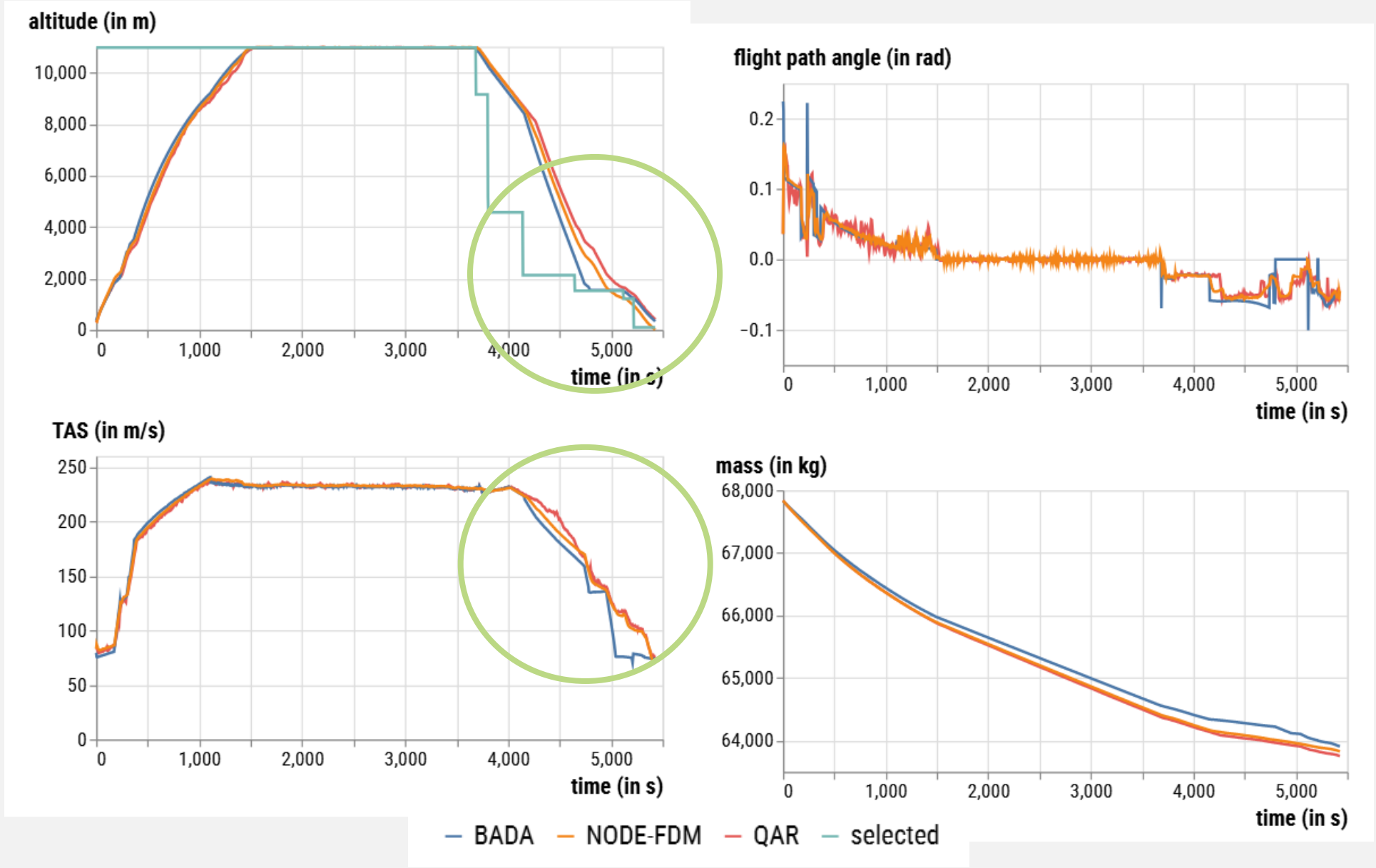
TABLE II: Comparison of NODE-FDM and BADA performance across phases. Metrics are reported as mean (standard deviation), based on 849,490 trajectory points across 500 flights.

parameter	phase	NODE-FDM			BADA		
		MAE	MAPE (%)	ME	MAE	MAPE (%)	ME
altitude [m]	All phases	61.90 (145.54)	2.49 (333.62)	0.47 (158.15)	167.47 (402.75)	5.03 (315.60)	-75.45 (429.61)
	Climb	93.03 (121.98)	3.36 (569.52)	57.64 (142.16)	214.69 (180.19)	6.14 (269.86)	175.79 (218.31)
	Level Flight	8.70 (33.50)	0.26 (6.16)	-1.51 (34.57)	12.32 (67.80)	0.28 (9.20)	-8.26 (68.42)
	Descent	182.04 (246.23)	7.95 (467.70)	-49.49 (302.19)	560.15 (713.73)	17.39 (637.75)	-509.53 (750.69)
true air speed [m/s]	All phases	1.35 (2.03)	0.75 (1.27)	-0.04 (2.44)	2.73 (6.16)	1.64 (5.18)	-1.34 (6.61)
	Climb	1.70 (1.93)	0.95 (1.27)	0.87 (2.42)	3.31 (3.33)	2.02 (2.66)	1.86 (4.31)
	Level Flight	0.70 (0.75)	0.32 (0.43)	-0.04 (1.03)	0.90 (1.62)	0.45 (1.41)	-0.62 (1.75)
	Descent	2.82 (3.33)	1.76 (2.02)	-0.92 (4.27)	7.32 (11.55)	4.64 (10.15)	-6.51 (12.02)
γ [deg]	All phases	0.24 (0.42)	N/A	0.05 (0.48)	0.46 (1.01)	N/A	0.05 (1.11)
	Climb	0.42 (0.51)	N/A	0.04 (0.65)	0.64 (1.11)	N/A	0.03 (1.28)
	Level Flight	0.08 (0.16)	N/A	0.03 (0.18)	0.07 (0.29)	N/A	0.03 (0.30)
	Descent	0.51 (0.60)	N/A	0.09 (0.78)	1.36 (1.49)	N/A	0.09 (2.01)
mass [kg]	All phases	64.89 (70.44)	0.10 (0.11)	7.26 (95.50)	154.89 (122.70)	0.25 (0.20)	152.90 (125.17)
	Climb	22.98 (31.34)	0.04 (0.05)	7.12 (38.21)	60.22 (58.35)	0.09 (0.09)	59.65 (58.94)
	Level Flight	74.91 (73.32)	0.12 (0.12)	6.06 (104.65)	170.43 (119.16)	0.27 (0.20)	168.63 (121.69)
	Descent	77.29 (74.27)	0.13 (0.12)	10.80 (106.64)	202.96 (130.73)	0.34 (0.22)	199.05 (136.61)

TABLE III: Comparison of consumption prediction performance (Predicted vs. BADA). Metrics are reported as mean (standard deviation), based on 849,490 trajectory points across 500 flights.

parameter	NODE-FDM			BADA		
	MAE [kg]	MAPE (%)	ME [kg]	MAE [kg]	MAPE (%)	ME [kg]
consumption	83.95 (79.65)	1.54 (1.13)	3.01 (115.74)	163.52 (129.68)	3.03 (1.75)	-149.67 (145.47)

Flight Example



Outlines

Can Neural ODEs be trained to learn accurate and physically consistent aircraft dynamics using QAR data?

1) Neural Ordinary Differential Equations
2) Data & Model
3) Baseline & Results
4) Limitations & Conclusions

Limitations and Future Work

- ✈️ Limited Dynamics → Include horizontal path
- 🧩 Partial Physical Constraints → physics-based loss regularization
- 📉 Dependence on QAR Data → Open ADS-B version and extended data
- 🧭 No Intent Prediction Yet → Predict intent to fill missing ADS-B data or predict from fligthplan

Conclusions

- **node-fdm** demonstates promising improvements on **trajectory reconstruction** compared with BADA particularly during descent phases.
- **High-fidelity performance is confirmed** on A320 QAR data across climb, cruise, and descent.
- **The hybrid physics–ML architecture shows strong potential**, yielding smoother and more coherent trajectories than purely data-driven approaches.
- Model and pipelines are now available into **open source github repository**:

https://github.com/eurocontrol-asu/node-fdm

SUPPORTING
EUROPEAN
AVIATION

SCAN
ME

Thank you !

{gabriel.jarry, ramon.dalmau, philippe.very}@eurocontrol.int

xavier.olive@onera.fr

node-fdm

ONERA
THE FRENCH AEROSPACE LAB

CONSORTIUM COORDINATOR sesar DEPLOYMENT MANAGER

FOUNDING MEMBER sesar JOINT UNDERTAKING

NETWORK MANAGER