REPRESENTATION–PROPAGATION LEARNING FOR CLOSED-LOOP ERROR-CONTROLLED LONG-HORIZON AIRCRAFT TRAJECTORY FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon aircraft trajectory forecasting is challenging because uncertainty in future flight intent can be further amplified during physical rollout. We propose Representation–Propagation Learning (RPL), a forecasting framework that explicitly separates future-intent representation from trajectory propagation. RPL first infers a latent future intent and decodes it into a reference trajectory, which is then realized through differentiable aircraft dynamics with bounded closed-loop feedback. Our main theoretical result characterizes the sensitivity of this propagation process. By separating feedback-stabilized tracking variables from the unavoidable position integrators, we show that, under finite-gain closed-loop tracking dynamics, perturbations in future intent induce a terminal-state sensitivity that grows at most as $O(\sqrt{T})$ with the forecast horizon, rather than exponentially. This provides a theoretical basis for controlling error amplification in long-horizon physical rollout without requiring contraction of the complete aircraft state. Experiments conducted on 21,816 held-out ADS-B forecasting windows demonstrate that RPL improves long-horizon forecasting accuracy and disturbance robustness in our evaluation. Although CFM performs better over short horizons, RPL's advantages become apparent as the prediction horizon extends; at a 120-second prediction horizon, RPL reduces ADE and FDE by 9.79% and 5.17%, respectively, compared with CFM. Controlled perturbation experiments further show that the closed-loop feedback mechanism substantially reduces error accumulation and performance degradation during rollout, supporting RPL's robustness in long-horizon trajectory prediction. You can get the code from this anonymous repository.

1 INTRODUCTION

Predicting future aircraft trajectories from partial observations is fundamental to air traffic management, autonomous aviation, and safety-critical decision making. Aircraft trajectory prediction has evolved from physics-informed and operational parameter estimation for climb prediction (Alligier et al., 2013; Alligier & Gianazza, 2018) to recurrent and hybrid deep-learning predictors (Pang et al., 2019; Zeng et al., 2020; Ma & Tian, 2020; Schimpf et al., 2023), Transformer-based models (Hao et al., 2024; Prutsch et al., 2026), and increasingly multimodal or generative formulations (Wu et al., 2022; Zhang & Liu, 2024; 2025; Yang et al., 2025; Figuet et al., 2026). Related work has also examined interaction-aware prediction (Xu et al., 2021; Patrikar et al., 2022), weather uncertainty (Pang et al., 2021), and prediction under degraded ADS-B observations (Yang et al., 2023).

Despite these advances, long-horizon prediction remains challenging because two sources of error interact: future motion must first be inferred from limited observations, and the inferred hypothesis must then be propagated over an extended horizon. Many existing forecasting models learn these two processes jointly through a direct mapping from observation history H_t to future trajectory $\hat{X}_{1:T}$. In recurrent or autoregressive prediction, rollout mismatch and exposure bias can cause local prediction errors to accumulate over time (Bengio et al., 2015). More recent work on long-horizon learned dynamics has likewise emphasized full-horizon rollout, error propagation, and stability-aware training (Rao et al., 2024; Lin et al., 2026; Li & Wan, 2026; Bachar, 2026). However, when future representation and trajectory propagation are learned as a single predictive mapping, errors in the inferred future hypothesis and errors introduced during rollout remain entangled. Consequently,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

it is difficult to determine whether long-horizon degradation originates from an inaccurate future representation or from additional sensitivity introduced while that representation is propagated forward.

We address this issue with Representation–Propagation Learning (RPL), which explicitly separates future-intent representation from physical trajectory propagation. Given an observation history H_t, RPL first infers a trajectory-level latent intent and decodes it into a reference trajectory $X^{\mathrm{ref}}_{1:T}$. Conditioned on the current state X_t, this reference is then realized as the final prediction $\hat{X}_{1:T}$ through a differentiable reduced-order aircraft model with bounded control and closed-loop feedback. This explicit interface makes representation quality and propagation sensitivity separately observable. It also allows the propagation module to correct deviations between the predicted reference and dynamically realizable aircraft motion during rollout.

RPL lies at the intersection of learned dynamics, dynamically feasible forecasting, and differentiable control. Continuous-time learning frameworks such as Neural ODEs, Latent ODEs, and Neural CDEs learn state evolution from data (Chen et al., 2018; Rubanova et al., 2019; Kidger et al., 2020), while Hamiltonian, symplectic, and universal differential-equation formulations introduce structural or physical inductive biases into learned dynamics (Greydanus et al., 2019; Zhong et al., 2020; Rackauckas et al., 2020). In the aircraft domain, Neural ODE formulations have also been used to model flight dynamics directly from data (Jarry et al., 2025). RPL differs from these approaches in that its primary objective is not to identify a complete learned dynamical system. Instead, it first represents a future-motion hypothesis explicitly and then studies the additional sensitivity introduced when this hypothesis is physically realized.

RPL is also related to dynamically feasible trajectory forecasting (Salzmann et al., 2020), differentiable prediction–control interfaces (Amos et al., 2018; Belbute-Peres et al., 2018; Drgoňa et al., 2022), and incremental or contraction-based analyses of closed-loop dynamical systems (Angeli, 2002; Lohmiller & Slotine, 1998; Tsukamoto & Chung, 2021a;b; Tsukamoto et al., 2021; Li et al., 2025). However, these lines of work do not explicitly isolate errors in a learned future representation from the additional sensitivity introduced by long-horizon physical propagation. Our objective therefore differs from directly optimizing a control policy, learning the complete system dynamics, or establishing contraction of the full aircraft state. Instead, RPL introduces an explicit interface between future representation and physical realization, allowing representation error and propagation-induced sensitivity to be analyzed separately.

A central question is whether repeated physical propagation amplifies small perturbations as the forecast horizon grows. We decompose the aircraft state into position variables and feedback-regulated tracking variables and analyze the incremental response of the tracking subsystem. Under a finite incremental-gain condition, the induced gain from perturbations entering the propagation module to tracking-error energy remains bounded independently of the forecast horizon. For fixed-energy input perturbations, accumulating this bounded tracking response through the position integrator yields a propagation-induced terminal sensitivity that grows at most as $O(\sqrt{T})$, rather than acquiring an additional exponential factor during rollout.

Importantly, this result characterizes the sensitivity introduced by the propagation mechanism; it does not imply that total forecasting error must scale as $O(\sqrt{T})$. The quality of the inferred future representation and the perturbation energy induced by representation error may themselves depend on the prediction horizon. The purpose of the analysis is therefore to isolate the additional contribution of physical propagation after a future hypothesis has already been formed.Our contributions are threefold:

- We introduce **Representation–Propagation Learning (RPL)**, a long-horizon forecasting framework that separates future-intent representation from controlled physical propagation and combines a learned reference trajectory with bounded closed-loop differentiable aircraft dynamics.

- We develop a **finite-gain analysis** that separates feedback-regulated tracking variables from position integrators. This yields a horizon-independent tracking gain and an $\mathcal{O}(\sqrt{T})$ bound on propagation-induced terminal sensitivity for fixed-energy perturbations, without requiring contraction of the complete aircraft state.

- We provide **theory-aligned empirical validation** on 21,816 held-out ADS-B forecasting windows. RPL becomes increasingly competitive as the forecast horizon grows and outperforms the best-performing baseline in our evaluation at 90–120 s. Controlled perturbation and ablation experiments further examine the finite-gain tracking mechanism, position-error accumulation, and the contribution of closed-loop realization.

2 Representation–Propagation Learning

2.1 RPL Overview

Given an observed trajectory history $H_t = (X_{t-H+1}, \ldots, X_t)$, RPL decomposes long-horizon forecasting into two stages: future-intent representation and controlled physical propagation. As illustrated in Fig. 1, the first stage predicts a reference trajectory representing a plausible future-motion hypothesis, while the second realizes this reference through bounded closed-loop correction and differentiable aircraft dynamics.

A history encoder G_θ produces a trajectory-level latent intent, which is decoded into a reference trajectory:

$$z \sim q_\theta(z \mid H_t), \qquad \tilde{X}^{\mathrm{ref}}_{1:T} = \Psi_\phi(z, X_t), \qquad X^{\mathrm{ref}}_{1:T} = \mathcal{S}\left(\tilde{X}^{\mathrm{ref}}_{1:T}\right), \tag{1}$$

where $\mathcal{S}$ is a differentiable smoothing operator that suppresses high-frequency variations in the predicted reference.

We write the aircraft state as $X_k = (P_k, Z_k)$, where $P_k = (x_k, y_k)$ denotes horizontal position and $Z_k = (h_k, v_k, \psi_k, v_{z,k})$ contains the variables directly regulated during propagation. The tracking error is compactly written as

$$e_k = Z^{\mathrm{ref}}_k - \hat{Z}_k, \tag{2}$$

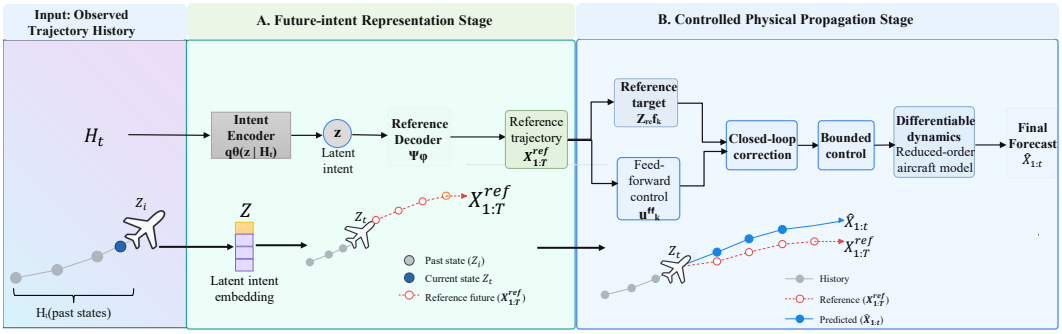

Figure 1: Overview of Representation–Propagation Learning (RPL).

2.2 Closed-Loop Physical Propagation

The reference trajectory determines a nominal feed-forward command u^{ff}_k. A feedback policy π_K produces a corrective command from the tracking error, whose contribution is modulated by an adaptive gate g_k. The executed command is

$$u_k = \Pi_{\mathcal{U}}\left(u^{\mathrm{ff}}_k + g_k \odot \pi_K(e_k)\right), \qquad g_k \in [0,1]^{d_u}, \tag{3}$$

where $\Pi_{\mathcal{U}}$ projects the command onto the admissible control set $\mathcal{U}$, ensuring bounded control throughout rollout.

The corrected command is propagated through a differentiable reduced-order aircraft model,

$$\hat{X}_{k+1} = F_{\Delta t}(\hat{X}_k, u_k), \qquad \hat{X}_0 = X_t, \tag{4}$$

yielding the final forecast $\hat{X}_{1:T}$. This closed-loop realization distinguishes RPL from directly treating the decoded reference as the final prediction: the propagation module can correct reference–dynamics mismatch while enforcing bounded physical execution.

For training, RPL jointly optimizes final-trajectory accuracy and reference quality, together with latent, control, and propagation regularization:

$$\mathcal{L} = \lambda_{\text{traj}}\mathcal{L}_{\text{traj}} + \lambda_{\text{ref}}\mathcal{L}_{\text{ref}} + \lambda_{\text{latent}}\mathcal{L}_{\text{latent}} + \lambda_{\text{stab}}\mathcal{L}_{\text{stab}} + \lambda_{\text{ctrl}}\mathcal{L}_{\text{ctrl}}. \tag{5}$$

The first two terms supervise the realized and reference trajectories, respectively; the remaining terms regularize the latent representation, perturbation amplification, and excessive feedback correction. Detailed architectures, control limits, and training hyperparameters are provided in the appendix.

The propagation stage defined by Eqs. (3)–(4) is the object whose perturbation sensitivity is analyzed in Section 3.

3 Theoretical Analysis

This section analyzes the sensitivity of the *propagation module* to perturbations in its reference-induced input. It asks whether a perturbation entering the closed-loop physical rollout acquires an additional exponential amplification as the rollout becomes longer.

3.1 Tracking–Position Decomposition

We decompose the aircraft state as

$$X_k = (P_k, Z_k), \qquad P_k = (x_k, y_k), \qquad Z_k = (h_k, v_k, \psi_k, v_{z,k}), \tag{6}$$

where P_k denotes horizontal position and Z_k collects the variables directly regulated by the closed-loop tracking system.

let $U = (U_0, \ldots, U_{T-1})$ denote the exogenous input sequence induced by a reference trajectory and supplied to the tracking subsystem. Here, U_k is an abstract reference-induced propagation input and should not be confused with the executed control command u_k used in the rollout model. Consider two reference hypotheses inducing input sequences U and V, and define

$$\Delta U_k = U_k - V_k, \qquad \Delta Z_k = Z_k^U - Z_k^V, \qquad \Delta P_k = P_k^U - P_k^V. \tag{7}$$

Within the bounded operating region of RPL, we assume that the continuous-time tracking dynamics $\dot{Z} = g(Z, U)$ admit a positive tracking-energy function $\mathcal{V}(\Delta Z)$ satisfying

$$\dot{\mathcal{V}}(\Delta Z) \le -\alpha\mathcal{V}(\Delta Z) + \beta\|\Delta U\|_2^2, \qquad \alpha > 0,\ \beta \ge 0. \tag{8}$$

The energy is locally equivalent to the squared tracking norm, i.e., $c_1\|\Delta Z\|_2^2 \le \mathcal{V}(\Delta Z) \le c_2\|\Delta Z\|_2^2$ for positive constants c_1, c_2 over the analyzed operating region.

Equation (8) states that tracking-state differences are dissipated by the closed-loop dynamics, while differences in the reference-induced input act as an external forcing term. Importantly, this condition is imposed only on the regulated tracking subsystem, not on the full aircraft state.

3.2 Finite-Gain Tracking Propagation

Integrating Eq. (8) gives

$$\mathcal{V}(t) \le e^{-\alpha t}\mathcal{V}(0) + \beta\int_0^t e^{-\alpha(t-s)}\|\Delta U(s)\|_2^2\, ds. \tag{9}$$

Thus, the tracking response is a stable convolution of the input perturbation.

The implemented rollout is evaluated at sampling interval Δt. Define the finite-horizon energy norms

$$\|\Delta Z\|_{Z,2} = \left(\Delta t\sum_{k=0}^{T-1}\|\Delta Z_k\|_2^2\right)^{1/2}, \qquad \|\Delta U\|_{U,2} = \left(\Delta t\sum_{k=0}^{T-1}\|\Delta U_k\|_2^2\right)^{1/2}. \tag{10}$$

Applying the energy bounds in Eq. (9) and discretizing at Δt yields

$$\|\Delta Z\|_{Z,2} \le C_{Z,0}\|\Delta Z_0\|_2 + C_{Z,U}\|\Delta U\|_{U,2}, \tag{11}$$

where $C_{Z,0}$ and $C_{Z,U}$ are finite constants independent of the forecast horizon T.

Equation (11) is the key finite-gain property: the induced gain from reference-input perturbation energy to tracking-error energy does not grow with the rollout horizon. Therefore, the tracking subsystem does not introduce an additional horizon-dependent exponential amplification.

3.3 Position Accumulation and Long-Horizon Sensitivity

We next characterize how the bounded tracking mismatch accumulates into horizontal position. Let the position dynamics be locally Lipschitz in the tracking state over the operating region, with constant L_g. Then

$$\|\Delta P_T\|_2 \le \|\Delta P_0\|_2 + L_g \Delta t \sum_{k=0}^{T-1} \|\Delta Z_k\|_2. \tag{12}$$

By Cauchy–Schwarz,

$$\Delta t \sum_{k=0}^{T-1} \|\Delta Z_k\|_2 \le \sqrt{T\Delta t}\, \|\Delta Z\|_{Z,2}. \tag{13}$$

Combining Eqs. (11)–(13) gives the following result.

Theorem 1 (Non-exponential propagation sensitivity) *Consider two trajectories propagated by the same bounded closed-loop RPL dynamics under reference-induced input sequences U and V. Assume that the tracking subsystem satisfies Eq.* (8) *and that the position dynamics are locally Lipschitz in the tracking variables. Then there exist finite constants $C_{X,0}$ and C_X, independent of T, such that*

$$\|\hat{X}_T^U - \hat{X}_T^V\|_2 \le C_{X,0}\|\Delta X_0\|_2 + C_X\sqrt{T\Delta t}\, \|\Delta U\|_{U,2}. \tag{14}$$

For trajectories initialized from the same observed state, $\Delta X_0 = 0$, and Eq. (14) reduces to

$$\|\hat{X}_T^U - \hat{X}_T^V\|_2 \le C_X\sqrt{T\Delta t}\, \|\Delta U\|_{U,2}. \tag{15}$$

Therefore, for fixed sampling interval and perturbations whose input-energy norm remains bounded independently of the horizon, the propagation-induced terminal sensitivity grows at most as $\mathcal{O}(\sqrt{T})$.

The square-root factor arises from accumulating a bounded tracking response through the position integrator; it is not an exponential instability term. Conversely, Eq. (14) does *not* state that nominal forecasting error must scale as $\mathcal{O}(\sqrt{T})$, since the quality of the inferred future representation and the perturbation energy $\|\Delta U\|_{U,2}$ may themselves depend on the prediction horizon. The theorem isolates only the additional sensitivity introduced by physical propagation. A sufficient matrix condition for Eq. (8), together with the detailed finite-gain and position-accumulation derivations, is provided in Appendix A.

4 Experiments

The experiments are organized to mirror the theoretical decomposition in Section 3. In particular, we examine four questions: **(1)** whether RPL remains accurate as the forecasting horizon increases; **(2)** whether separating reference representation from physical realization introduces controllable realization error; **(3)** whether the closed-loop tracking subsystem exhibits the finite-gain behavior described in Eq. (11); and **(4)** whether the accumulation of tracking perturbations into position is consistent with the $\sqrt{T}$ dependence in Eq. (14). The experiments do not constitute a proof of the theorem—the proof is given analytically in Section 3 and Appendix A—but directly test the mechanisms and operating conditions on which the theoretical result is based.

4.1 Data and Evaluation Protocol

The raw ADS-B data is sourced from TrajAir, with detailed data processing steps presented in the appendix. Each processed ADS-B window comprises 40 historical states and 120 future states. We evaluate prediction horizons of 30, 60, 90, and 120 seconds. The constructed training, validation, and test sets contain 78,308, 25,650, and 21,816 windows, respectively. All methods are evaluated on the same ordered set of test windows.

We compare RPL with constant velocity (CV), constant turn rate and velocity (CTRV), LSTM Seq2Seq Sutskever et al. (2014), Transformer Vaswani et al. (2017), PatchTST Nie et al. (2023), ASCENT-adapted Prutsch et al. (2026), CVAE-direct Sohn et al. (2015), and conditional flow matching (CFM)-adapted Lipman et al. (2023); Figuet et al. (2026). These neural baselines are independently trained task adaptations and do not share the RPL representation network.

Evaluation Metrics. Let $\mathbf{p}_{i,t}$ and $\hat{\mathbf{p}}_{i,t}$ denote the ground-truth and predicted horizontal positions of trajectory i at future step t, respectively, and define $d_{i,t} = \|\hat{\mathbf{p}}_{i,t} - \mathbf{p}_{i,t}\|_2$. For N test trajectories and K_H prediction steps within horizon H, we use

$$\mathrm{ADE@}H = \frac{1}{NK_H}\sum_{i=1}^{N}\sum_{t=1}^{K_H} d_{i,t}, \qquad \mathrm{FDE@}H = \frac{1}{N}\sum_{i=1}^{N} d_{i,K_H}. \tag{16}$$

Both are measured in metres, and lower values are better.

For neural models trained with $S = 3$ random seeds, results are reported as

$$\bar{M} = \frac{1}{S}\sum_{s=1}^{S} M_s, \qquad \sigma_M = \sqrt{\frac{1}{S-1}\sum_{s=1}^{S}(M_s - \bar{M})^2}. \tag{17}$$

The horizon-growth factor is

$$G_M(H_2, H_1) = \frac{M@H_2}{M@H_1}, \qquad M \in \{\mathrm{ADE}, \mathrm{FDE}\}. \tag{18}$$

For controlled perturbation experiments, let $\delta\mathbf{x}_t$, $\delta\mathbf{u}_t$, and $\delta\mathbf{p}_t$ denote deviations in the normalized tracking state, input, and horizontal position, respectively. We define

$$R_{\mathrm{trk}}(H) = \frac{\|\delta\mathbf{x}_{K_H}\|_2^2}{\|\delta\mathbf{x}_0\|_2^2}, \qquad G_{\mathrm{trk}}(H) = \frac{\left(\sum_{t=1}^{K_H}\|\delta\mathbf{x}_t\|_2^2\right)^{1/2}}{\left(\sum_{t=1}^{K_H}\|\delta\mathbf{u}_t\|_2^2\right)^{1/2}}. \tag{19}$$

The terminal-position gain normalized by rollout duration is

$$G_{\mathrm{pos}}(H) = \frac{\|\delta\mathbf{p}_{K_H}\|_2}{\sqrt{T_H}\left(\sum_{t=1}^{K_H}\|\delta\mathbf{u}_t\|_2^2\right)^{1/2}}. \tag{20}$$

Finally, for the local closed-loop Jacobian $\mathbf{J}_t = \partial\mathbf{F}_{\mathrm{cl}}/\partial\mathbf{x}|_{(\mathbf{x}_t,\mathbf{u}_t)}$, we use

$$\rho(\mathbf{J}_t) = \max_j |\lambda_j(\mathbf{J}_t)|, \qquad r_{\mathrm{Lyap}} = \lambda_{\max}\left(\mathbf{J}_t^\top\mathbf{P}\mathbf{J}_t - \mathbf{P}\right), \quad \mathbf{P} \succ 0. \tag{21}$$

These quantities are used as local diagnostics rather than as a global stability certificate.

4.2 Long-Horizon Forecasting Accuracy

The first experiment evaluates the practical motivation for the theoretical analysis: whether prediction quality degrades differently as the rollout horizon increases. The bound in Eq. (14) concerns propagation sensitivity rather than ground-truth forecasting error, so nominal ADE/FDE cannot by itself verify the theorem. Nevertheless, horizon-dependent accuracy provides an important empirical test of whether the proposed structured propagation remains useful as the rollout becomes longer.

Table 1: Horizontal ADE/FDE in metres on 21,816 test windows. Each horizon is evaluated as a prefix of the same 120 s prediction. Neural entries are mean ± sample SD over three training seeds. Lower is better.

Method	ADE@30	FDE@30	ADE@60	FDE@60	ADE@90	FDE@90	ADE@120	FDE@120
CV	118.00	282.47	337.87	851.24	635.41	1612.79	993.77	2510.12
CTRV	101.70	256.10	330.91	905.06	677.69	1834.46	1094.95	2818.61
LSTM-Seq2Seq	117.63±6.26	215.44±4.10	246.12±6.38	535.71±12.39	404.69±9.86	900.43±23.24	579.50±14.67	1307.33±32.64
Transformer	116.36±2.83	215.94±2.19	248.08±3.81	547.32±13.85	412.78±8.67	927.01±31.14	591.56±16.59	1314.58±50.99
PatchTST	156.57±6.60	256.81±4.72	313.55±6.37	706.61±6.36	546.37±7.03	1312.91±11.72	823.14±9.32	1974.49±20.41
ASCENT-adapted	140.65±15.87	317.07±41.08	363.54±51.47	863.09±123.77	636.62±83.92	1489.19±183.99	928.23±110.57	2097.67±219.79
CVAE-direct	**85.04**±1.79	198.92±4.38	229.33±1.60	552.62±3.34	406.49±2.54	959.66±15.93	599.62±7.26	1388.55±25.54
CFM-Aircraft-adapted	87.83±1.12	**195.27**±2.57	**221.53**±3.64	**517.20**±12.28	382.50±9.19	884.26±29.31	557.54±16.62	1271.87±48.66
RPL	106.75±0.51	244.84±1.55	263.22±2.36	583.20±7.36	**376.74**±5.25	**803.96**±14.27	**502.98**±7.99	**1206.17**±17.79

A clear horizon-dependent pattern appears in Table 1. CFM is stronger at short horizons: at 30–60 s it achieves lower ADE and FDE than RPL. The ordering reverses as the prediction horizon increases. At

90 s, RPL reaches 376.74 ± 5.25 m ADE and 803.96 ± 14.27 m FDE, compared with 382.50 ± 9.19 m and 884.26 ± 29.31 m for CFM. At 120 s, RPL further reduces ADE and FDE to 502.98 ± 7.99 m and 1206.17 ± 17.79 m, compared with 557.54 ± 16.62 m and 1271.87 ± 48.66 m for CFM.

From 30 to 120 s, the mean ADE increases by approximately $4.71\times$ for RPL and $6.35\times$ for CFM, while FDE increases by approximately $4.93\times$ and $6.51\times$, respectively. These ratios are descriptive rather than estimates of the theoretical gain, but they show that the relative advantage of RPL becomes more pronounced as the horizon increases.

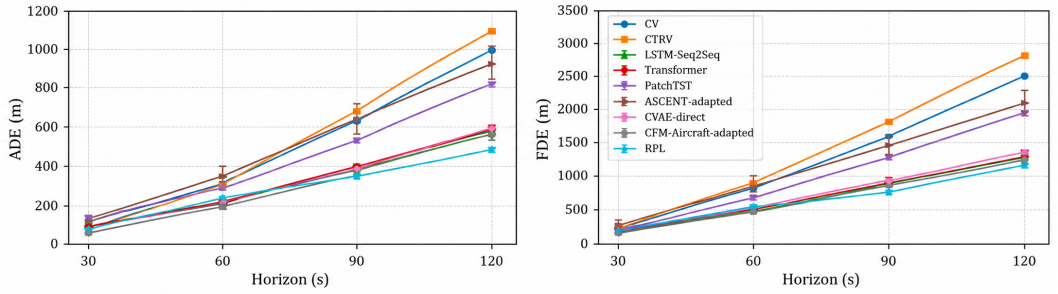

Figure 2: Prediction accuracy as a function of forecast horizon. CFM is stronger at short horizons, whereas RPL overtakes it at 90–120 s. This horizon-dependent crossover motivates the direct propagation-sensitivity tests that follow.

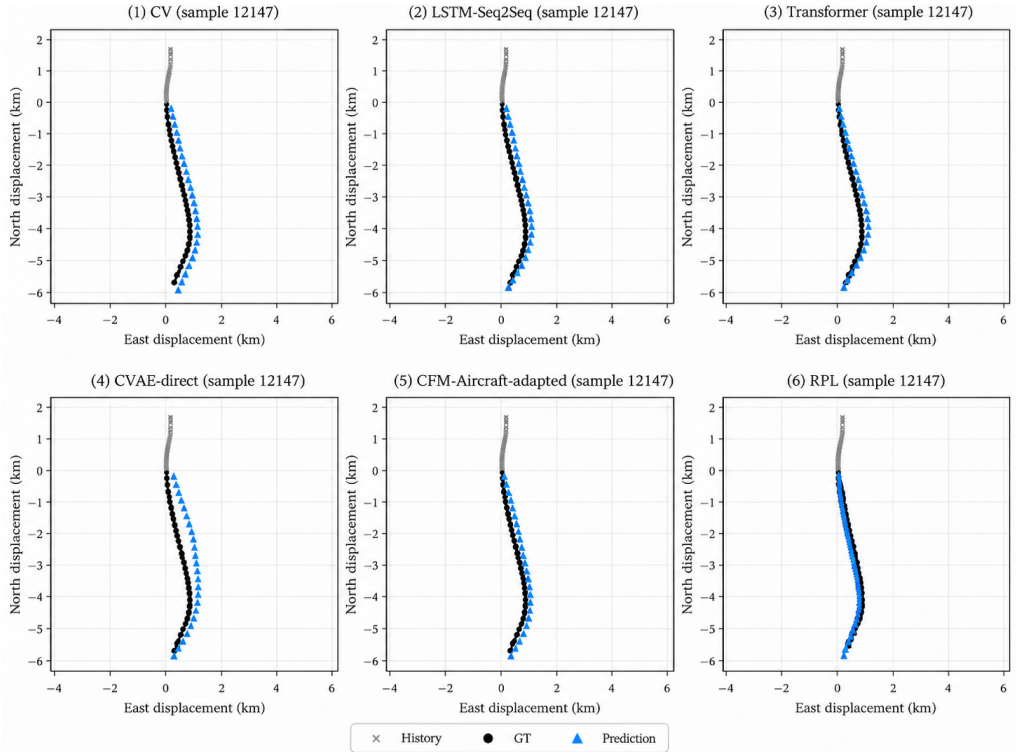

Figure 3: Representative trajectory predictions from RPL and selected baselines. Gray markers denote observed history, black markers ground truth, and colored markers predictions.Samples were selected randomly.

4.3 Ablation Experiments

To isolate errors from future-intent representation and physical realization, we evaluate three stages of the RPL pipeline: the frozen reference trajectory, its direct dynamics projection, and the final trajectory produced by the full realization module. Table 2 compares these stages together with component ablations across forecasting horizons.

Table 2: RPL realization and component ablations. Neural variants are reported as mean ± sample SD over three training seeds. Lower is better.

	30 s		60 s		90 s		120 s	
Variant	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
Frozen reference	118.89	243.66	257.18	548.37	419.41	934.52	601.52	1340.04
Reference projection	110.15	246.40	260.48	566.51	424.59	935.77	604.80	1350.47
w/o reference smoothing	128.55±0.20	282.78±1.41	300.99±2.09	652.65±5.69	479.77±4.05	1006.64±11.22	655.93±7.08	1359.63±23.42
w/o adaptive correction gate	106.90±0.29	245.17±0.80	263.54±1.10	584.71±3.53	431.49±2.92	942.35±10.79	607.89±6.45	1326.26±24.05
w/o realization pretraining/loss	107.18±0.53	246.73±1.57	268.39±1.11	603.05±2.23	442.83±1.69	968.73±5.49	622.63±3.22	1350.79±10.60
RPL (Full)	106.75±0.51	244.84±1.55	263.22±2.36	583.20±7.36	**376.74**±5.25	**803.96**±14.27	**502.98**±7.99	**1206.17**±17.79

The benefit of learned realization becomes most evident at long horizons. At 120 s, the frozen reference and direct dynamics projection achieve 601.52/1340.04 m and 604.80/1350.47 m ADE/FDE, respectively, whereas full RPL reduces the errors to 502.98 ± 7.99 m and 1206.17 ± 17.79 m. Thus, relative to direct dynamics projection, the full realization module improves ADE and FDE by 101.82 m and 144.30 m, indicating that its gain comes from learned closed-loop realization rather than dynamics projection alone.

The component ablations further show that reference smoothing and explicit realization training are particularly important for long-horizon performance. Removing them increases 120 s ADE/FDE to 655.93/1359.63 m and 622.63/1350.79 m, respectively, while removing the adaptive correction gate yields 607.89/1326.26 m. Overall, the larger differences at 90–120 s support the role of structured realization in limiting long-horizon rollout degradation.

4.4 Closed-Loop Robustness and Theory-Aligned Diagnostics

We next examine whether the empirical behavior of RPL is consistent with the finite-gain propagation mechanism analyzed in Section 3. Identical disturbances are injected into paired open- and closed-loop rollouts while keeping the nominal trajectory fixed, so that the effect of feedback can be isolated.

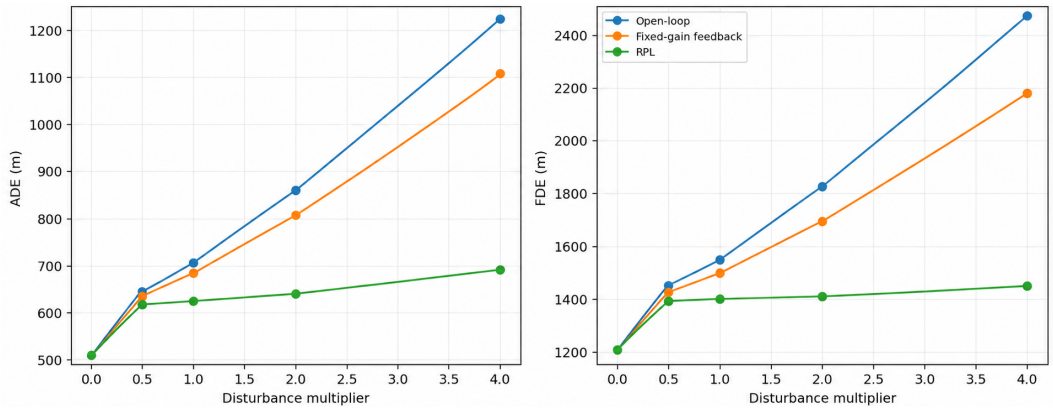

Figure 4: Paired open- and closed-loop forecasting errors under increasing disturbance severity. Learned RPL feedback substantially limits disturbance-induced error growth.

At disturbance multiplier 1, open-loop ADE/FDE increase to 704.80/1549.71 m, whereas RPL feedback reduces them to 624.93/1397.25 m. At the largest tested multiplier 4, the gap becomes more pronounced: open-loop propagation reaches 1224.94/2492.02 m, while RPL remains at

692.88/1449.09 m. These results show that closed-loop feedback substantially limits the additional degradation introduced during physical rollout.

To more directly connect the experiments with the analysis in Section 3, we also evaluate controlled perturbations of the tracking subsystem. With identical future inputs and perturbed initial tracking states, the closed-loop tracking-energy ratio decreases to approximately 0.29 at 120 s, compared with 1.28 in open loop. Under matched input perturbations, the median normalized tracking gain is 1.39 in closed loop versus 5.36 in open loop. The corresponding terminal-position gain normalized by $\sqrt{\text{time}}$ is 3.88 and 39.97, respectively.

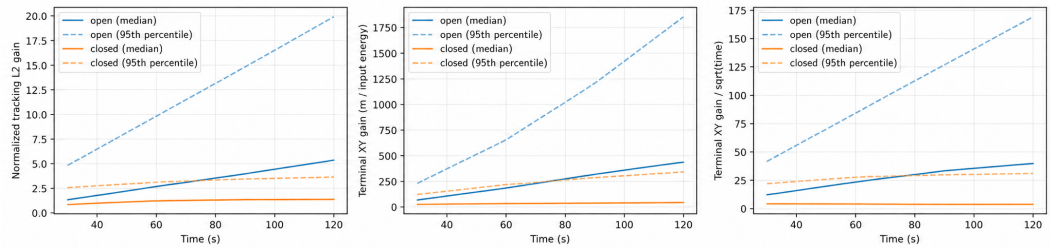

Figure 5: Theory-aligned perturbation diagnostics. Closed-loop tracking gain and the position gain normalized by $\sqrt{\text{time}}$ remain substantially below their open-loop counterparts as the forecast horizon increases.

These observations are consistent with the two mechanisms used in the theoretical analysis: a bounded input-to-tracking response and non-exponential accumulation of tracking perturbations through the position integrator. A sampled local Jacobian audit further gives a median spectral radius close to 0.991, although the strict diagonal-metric Lyapunov residual is not satisfied at all sampled operating points. We therefore interpret these experiments as empirical support for the finite-gain mechanism.

5 Conclusion and Limitations

We introduced Representation–Propagation Learning (RPL), which separates future-intent representation from controlled physical propagation for long-horizon aircraft trajectory forecasting. This decomposition makes it possible to distinguish errors in the predicted future hypothesis from sensitivity introduced during physical rollout. Empirically, RPL exhibits a clear horizon-dependent advantage: although CFM performs better at short horizons, RPL overtakes it at 90–120 s and, at 120 s, reduces ADE and FDE by 9.79% and 5.17%, respectively.

The robustness and ablation studies provide complementary evidence for the proposed mechanism. Closed-loop feedback substantially reduces disturbance-induced degradation relative to open-loop propagation, while controlled perturbation experiments show lower tracking and terminal-position gains. These observations are consistent with the finite-gain analysis and the $\sqrt{T}$-type accumulation of bounded tracking perturbations through the position integrator. Component ablations further show that reference smoothing and explicit realization training are particularly important for long-horizon performance, while the adaptive correction gate provides an additional measurable benefit.

Several limitations remain. RPL does not explicitly incorporate future flight plans or weather conditions, and its use of absolute spatial context may limit transfer across airspaces. Moreover, the injected disturbances are controlled stress tests rather than calibrated models of operational atmospheric uncertainty. The local Jacobian diagnostics are evaluated only at sampled operating points and therefore provide empirical support for the finite-gain mechanism rather than a global contraction certificate.

Overall, the results suggest that explicitly separating future representation from physical realization is useful when propagation-induced error becomes increasingly important over long horizons. By combining a learned reference trajectory with bounded closed-loop differentiable dynamics, RPL provides a framework in which long-horizon forecasting accuracy and propagation sensitivity can be studied and controlled within the same model.

AI USE STATEMENT

We utilized LLMs to translate and polish the text, assist with the derivation of certain formulas, and aid in locating and reproducing the paper on the "Ascent" method; however, the core ideas, experimental setup, and implementation were entirely carried out by us.

REFERENCES

B. Amos, I. D. J. Rodriguez, J. Sacks, B. Boots, and J. Z. Kolter. Differentiable MPC for end-to-end planning and control. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pp. 8299–8310, 2018.

D. Angeli. A Lyapunov approach to incremental stability properties. *IEEE Transactions on Automatic Control*, 47(3):410–421, 2002. doi:10.1109/9.989067.

F. de A. Belbute-Peres, K. A. Smith, K. R. Allen, J. B. Tenenbaum, and J. Z. Kolter. End-to-end differentiable physics for learning and control. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pp. 7178–7189, 2018.

J. Drgoňa, K. Kiš, A. Tuor, D. Vrabie, and M. Klaučo. Differentiable predictive control: Deep learning alternative to explicit model predictive control for unknown nonlinear systems. *Journal of Process Control*, 116:80–92, 2022. doi:10.1016/j.jprocont.2022.06.001.

B. Figuet, T. Krauth, and S. Barry. Generative short-term aircraft trajectory prediction with conditional flow matching. *Journal of Open Aviation Science*, 4(2), 2026. doi:10.59490/joas.2026.8468.

H. Hao, X. Si, M. Li, and J. Liang. Flight trajectory change prediction via patched spatial-temporal transformer. In *2024 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 10394–10398, 2024. doi:10.1109/IGARSS53475.2024.10641236.

G. Jarry, R. Dalmau, X. Olive, and P. Very. A neural ODE approach to aircraft flight dynamics modelling. In *Proceedings of the 15th SESAR Innovation Days*, Bled, Slovenia, 2025.

H. Li, X. Zhong, B. Hu, and H. Zhang. Neural contraction metrics with formal guarantees for discrete-time nonlinear dynamical systems. In *Proceedings of the 7th Annual Learning for Dynamics and Control Conference (L4DC)*, volume 283 of *Proceedings of Machine Learning Research*, pp. 1447–1459, 2025.

Q. Li and M. Wan. Stability-regularized residual neural ODEs: From rollout-error contraction diagnostics to a train-time method for robust long-horizon forecasting. *Mathematics*, 14(15):2850, 2026. doi:10.3390/math14152850.

H. Lin, S. Xiao, Y.-C. Li, Z. Zhang, Y. Sun, C. Jia, and Y. Yu. ADM-v2: Pursuing full-horizon roll-out in dynamics models for offline policy learning and evaluation. In *International Conference on Learning Representations (ICLR)*, 2026.

Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

W. Lohmiller and J.-J. E. Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, 1998. doi:10.1016/S0005-1098(98)00019-3.

Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations (ICLR)*, 2023.

Y. Pang, N. Xu, and Y. Liu. Aircraft trajectory prediction using LSTM neural network with embedded convolutional layer. *Annual Conference of the PHM Society*, 11(1), 2019. doi:10.36001/phmconf.2019.v11i1.849.

J. Patrikar, B. G. Moon, J. Oh, and S. A. Scherer. Predicting like a pilot: Dataset and method to predict socially-aware aircraft trajectories in non-towered terminal airspace. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2525–2531, 2022. doi:10.1109/ICRA46639.2022.9811972.

A. Prutsch, D. Schinagl, and H. Possegger. ASCENT: Transformer-based aircraft trajectory prediction in non-towered terminal airspace. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

P. P. Rao, A. Saviolo, T. Castiglione Ferrari, and G. Loianno. Learning long-horizon predictions for quadrotor dynamics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12758–12765, 2024. doi:10.1109/IROS58592.2024.10801793.

T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision (ECCV)*, pp. 683–700, 2020. doi:10.1007/978-3-030-58523-5_40.

K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3483–3491, 2015.

R. Sun, H. Zang, X. Li, and R. Islam. Learning latent dynamic robust representations for world models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 47234–47260, 2024.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

H. Tsukamoto and S.-J. Chung. Learning-based robust motion planning with guaranteed stability: A contraction theory approach. *IEEE Robotics and Automation Letters*, 6(4):6164–6171, 2021. doi:10.1109/LRA.2021.3091019.

H. Tsukamoto and S.-J. Chung. Neural contraction metrics for robust estimation and control: A convex optimization approach. *IEEE Control Systems Letters*, 5(1):211–216, 2021. doi:10.1109/LCSYS.2020.3001646.

H. Tsukamoto, S.-J. Chung, J.-J. E. Slotine, and C. Fan. A theoretical overview of neural contraction metrics for learning-based control with guaranteed stability. In *60th IEEE Conference on Decision and Control (CDC)*, pp. 2949–2954, 2021. doi:10.1109/CDC45484.2021.9682859.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

S. Yang, L. Liu, B. Chen, S. Cheng, Z. Shi, and Z. Zou. GooDFlight: Goal-oriented diffusion model for flight trajectory prediction. *IEEE Transactions on Aerospace and Electronic Systems*, 61(3):7447–7465, 2025. doi:10.1109/TAES.2025.3536436.

R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pp. 6571–6583, 2018.

Y. Rubanova, R. T. Q. Chen, and D. Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 5320–5330, 2019.

P. Kidger, J. Morrill, J. Foster, and T. Lyons. Neural controlled differential equations for irregular time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

S. Greydanus, M. Dzamba, and J. Yosinski. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 15379–15389, 2019.

Y. D. Zhong, B. Dey, and A. Chakraborty. Symplectic ODE-Net: Learning Hamiltonian dynamics with control. In *International Conference on Learning Representations (ICLR)*, 2020.

C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, A. Ramadhan, and A. Edelman. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*, 2020.

S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015.

M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm. Bringing up OpenSky: A large-scale ADS-B sensor network for research. In *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks (IPSN)*, pp. 83–94, 2014. doi:10.1109/IPSN.2014.6846743.

R. Alligier, D. Gianazza, and N. Durand. Learning the aircraft mass and thrust to improve the ground-based trajectory prediction of climbing flights. *Transportation Research Part C: Emerging Technologies*, 36:45–60, 2013. doi:10.1016/j.trc.2013.08.006.

R. Alligier and D. Gianazza. Learning aircraft operational factors to improve aircraft climb prediction: A large scale multi-airport study. *Transportation Research Part C: Emerging Technologies*, 96:72–95, 2018. doi:10.1016/j.trc.2018.08.012.

W. Zeng, Z. Quan, Z. Zhao, C. Xie, and X. Lu. A deep learning approach for aircraft trajectory prediction in terminal airspace. *IEEE Access*, 8:151250–151266, 2020. doi:10.1109/ACCESS.2020.3016289.

L. Ma and S. Tian. A hybrid CNN-LSTM model for aircraft 4D trajectory prediction. *IEEE Access*, 8:134668–134680, 2020. doi:10.1109/ACCESS.2020.3010963.

Z. Xu, W. Zeng, X. Chu, and P. Cao. Multi-aircraft trajectory collaborative prediction based on social long short-term memory network. *Aerospace*, 8(4):115, 2021. doi:10.3390/aerospace8040115.

Y. Pang, X. Zhao, H. Yan, and Y. Liu. Data-driven trajectory prediction with weather uncertainties: A Bayesian deep learning approach. *Transportation Research Part C: Emerging Technologies*, 130:103326, 2021. doi:10.1016/j.trc.2021.103326.

X. Wu, H. Yang, H. Chen, Q. Hu, and H. Hu. Long-term 4D trajectory prediction using generative adversarial networks. *Transportation Research Part C: Emerging Technologies*, 136:103554, 2022. doi:10.1016/j.trc.2022.103554.

Z. Yang, X. Kang, Y. Gong, and J. Wang. Aircraft trajectory prediction and aviation safety in ADS-B failure conditions based on neural network. *Scientific Reports*, 13:19677, 2023. doi:10.1038/s41598-023-46914-2.

N. Schimpf, Z. Wang, S. Li, E. J. Knoblock, H. Li, and R. D. Apaza. A generalized approach to aircraft trajectory prediction via supervised deep learning. *IEEE Access*, 11:116183–116195, 2023. doi:10.1109/ACCESS.2023.3325053.

H. Zhang and Z. Liu. Four-dimensional aircraft trajectory prediction based on generative deep learning. *Journal of Aerospace Information Systems*, 21(7):554–567, 2024. doi:10.2514/1.I011333.

H. Zhang and Z. Liu. Four-dimensional aircraft trajectory prediction with a generative deep learning and clustering approach. *Journal of Aerospace Information Systems*, 22(2):90–102, 2025. doi:10.2514/1.I011454.

M. Bachar. Deep learning-based residual augmentation of neural ODE approximations: Rollout error propagation, contraction diagnostics, and CRN case study. *Mathematics*, 14(12):2147, 2026. doi:10.3390/math14122147.

A Detailed Stability Derivations

This appendix provides the detailed derivations underlying the non-exponential propagation-sensitivity result in Section 3. The analysis concerns the sensitivity introduced by the propagation module and does not assume that the inferred future representation itself is error free.

A.1 Matrix Sufficient Condition for Tracking Dissipativity

Consider the continuous-time closed-loop tracking dynamics

$$\dot{Z} = g(Z, U), \tag{22}$$

where Z denotes the feedback-regulated tracking variables and U is the reference-induced propagation input.

For two trajectories, define

$$\Delta Z = Z^U - Z^V, \qquad \Delta U = U - V. \tag{23}$$

By the mean-value theorem, their incremental dynamics can be written as

$$\Delta \dot{Z} = \bar{A}_{\mathrm{cl}} \Delta Z + \bar{B}_{\mathrm{cl}} \Delta U, \tag{24}$$

where

$$A_{\mathrm{cl}}(Z, U) = \frac{\partial g}{\partial Z}, \qquad B_{\mathrm{cl}}(Z, U) = \frac{\partial g}{\partial U}, \tag{25}$$

and the barred matrices denote averages along the line segment connecting the two state–input pairs.

Let

$$\mathcal{V}(\Delta Z) = \Delta Z^\top P \Delta Z, \qquad P \succ 0. \tag{26}$$

Assume that throughout the bounded operating region,

$$A_{\mathrm{cl}}^\top P + P A_{\mathrm{cl}} \preceq -2\alpha_0 P, \qquad \|P B_{\mathrm{cl}}\|_2 \le b, \tag{27}$$

for finite constants $\alpha_0 > 0$ and $b \ge 0$.

Differentiating $\mathcal{V}$ along Eq. (24) gives

$$\begin{aligned} \dot{\mathcal{V}} &= \Delta Z^\top \left(\bar{A}_{\mathrm{cl}}^\top P + P \bar{A}_{\mathrm{cl}} \right) \Delta Z + 2 \Delta Z^\top P \bar{B}_{\mathrm{cl}} \Delta U \\ &\le -2\alpha_0 \mathcal{V} + 2b \|\Delta Z\|_2 \|\Delta U\|_2. \end{aligned} \tag{28}$$

Because

$$\mathcal{V} \ge \lambda_{\min}(P) \|\Delta Z\|_2^2, \tag{29}$$

Young's inequality implies

$$\begin{aligned} 2b \|\Delta Z\|_2 \|\Delta U\|_2 &\le \alpha_0 \lambda_{\min}(P) \|\Delta Z\|_2^2 + \frac{b^2}{\alpha_0 \lambda_{\min}(P)} \|\Delta U\|_2^2 \\ &\le \alpha_0 \mathcal{V} + \frac{b^2}{\alpha_0 \lambda_{\min}(P)} \|\Delta U\|_2^2. \end{aligned} \tag{30}$$

Substituting this result into Eq. (28) yields

$$\dot{\mathcal{V}} \le -\alpha_0 \mathcal{V} + \frac{b^2}{\alpha_0 \lambda_{\min}(P)} \|\Delta U\|_2^2. \tag{31}$$

Thus the dissipativity inequality used in the main text is satisfied with

$$\alpha = \alpha_0, \qquad \beta = \frac{b^2}{\alpha_0 \lambda_{\min}(P)}. \tag{32}$$

Equation (27) is a sufficient rather than necessary condition. Consequently, the sampled Jacobian experiment in Section 4.4 is interpreted as a local diagnostic and not as a global contraction certificate.

A.2 Finite-Gain Tracking Propagation

Starting from

$$\dot{\mathcal{V}}(t) \le -\alpha \mathcal{V}(t) + \beta \|\Delta U(t)\|_2^2, \tag{33}$$

multiplying by $e^{\alpha t}$ and integrating gives

$$\mathcal{V}(t) \le e^{-\alpha t}\mathcal{V}(0) + \beta \int_0^t e^{-\alpha(t-s)} \|\Delta U(s)\|_2^2 \, ds. \tag{34}$$

Suppose

$$c_1 \|\Delta Z\|_2^2 \le \mathcal{V}(\Delta Z) \le c_2 \|\Delta Z\|_2^2, \qquad c_1, c_2 > 0. \tag{35}$$

Integrating Eq. (33) over $[0, \mathcal{T}]$ gives

$$\mathcal{V}(\mathcal{T}) - \mathcal{V}(0) \le -\alpha \int_0^{\mathcal{T}} \mathcal{V}(t)\, dt + \beta \int_0^{\mathcal{T}} \|\Delta U(t)\|_2^2 \, dt. \tag{36}$$

Since $\mathcal{V}(\mathcal{T}) \ge 0$,

$$\alpha \int_0^{\mathcal{T}} \mathcal{V}(t)\, dt \le \mathcal{V}(0) + \beta \int_0^{\mathcal{T}} \|\Delta U(t)\|_2^2 \, dt. \tag{37}$$

Using Eq. (35),

$$\int_0^{\mathcal{T}} \|\Delta Z(t)\|_2^2 \, dt \le \frac{c_2}{\alpha c_1} \|\Delta Z_0\|_2^2 + \frac{\beta}{\alpha c_1} \int_0^{\mathcal{T}} \|\Delta U(t)\|_2^2 \, dt. \tag{38}$$

Therefore,

$$\|\Delta Z\|_{L_2} \le \sqrt{\frac{c_2}{\alpha c_1}} \|\Delta Z_0\|_2 + \sqrt{\frac{\beta}{\alpha c_1}} \|\Delta U\|_{L_2}. \tag{39}$$

For the sampled implementation, define

$$\|\Delta Z\|_{Z,2} = \left(\Delta t \sum_{k=0}^{T-1} \|\Delta Z_k\|_2^2 \right)^{1/2}, \tag{40}$$

and

$$\|\Delta U\|_{U,2} = \left(\Delta t \sum_{k=0}^{T-1} \|\Delta U_k\|_2^2 \right)^{1/2}. \tag{41}$$

The sampled finite-gain relation therefore takes the form

$$\|\Delta Z\|_{Z,2} \le C_{Z,0} \|\Delta Z_0\|_2 + C_{Z,U} \|\Delta U\|_{U,2}, \tag{42}$$

where $C_{Z,0}$ and $C_{Z,U}$ are finite and do not depend on the forecast length T.

This is the key horizon-independent tracking-gain property: increasing the rollout horizon does not introduce an additional exponential factor into the input-to-tracking response.

A.3 Position Accumulation for Equal Initial Conditions

Let

$$P = (x, y) \tag{43}$$

denote horizontal position. Its reduced-order dynamics are

$$\dot{P} = g_P(Z) = \begin{bmatrix} v \cos\psi \\ v \sin\psi \end{bmatrix}. \tag{44}$$

On the bounded operating region, assume that g_P is locally Lipschitz in Z:

$$\|g_P(Z^U) - g_P(Z^V)\|_2 \le L_g \|\Delta Z\|_2. \tag{45}$$

Then

$$\|\Delta P_{\mathcal{T}}\|_2 \le \|\Delta P_0\|_2 + L_g \int_0^{\mathcal{T}} \|\Delta Z(t)\|_2 \, dt. \tag{46}$$

First consider two rollouts initialized from the same observed state, so that

$$\Delta P_0 = 0, \qquad \Delta Z_0 = 0. \tag{47}$$

By Cauchy–Schwarz,

$$\int_0^{\mathcal{T}} \|\Delta Z(t)\|_2 \, dt \le \sqrt{\mathcal{T}} \left(\int_0^{\mathcal{T}} \|\Delta Z(t)\|_2^2 \, dt \right)^{1/2}. \tag{48}$$

Combining this inequality with Eq. (39) gives

$$\|\Delta P_{\mathcal{T}}\|_2 \le L_g \sqrt{\frac{\beta}{\alpha c_1}} \sqrt{\mathcal{T}} \|\Delta U\|_{L_2}. \tag{49}$$

In discrete time, $\mathcal{T} = T\Delta t$, and hence

$$\|\Delta P_T\|_2 \le C_P \sqrt{T\Delta t} \|\Delta U\|_{U,2}, \tag{50}$$

for a finite constant C_P independent of T.

The terminal tracking-state difference is also bounded. From Eq. (34) with $\Delta Z_0 = 0$,

$$\mathcal{V}(\mathcal{T}) \le \beta \|\Delta U\|_{L_2}^2, \tag{51}$$

so

$$\|\Delta Z_{\mathcal{T}}\|_2 \le \sqrt{\frac{\beta}{c_1}} \|\Delta U\|_{L_2}. \tag{52}$$

For a fixed sampling interval this horizon-independent terminal term can be absorbed into the coefficient multiplying $\sqrt{T\Delta t}$.

A.4 Initial-State Contribution and Proof of Theorem 1

We now consider different initial conditions as well as different reference-induced inputs.

Introduce an intermediate trajectory $\widetilde{X}$ that starts from the first trajectory's initial state X_0^U but is propagated using the second input sequence V. By the triangle inequality,

$$\|\hat{X}_T^U - \hat{X}_T^V\|_2 \le \|\hat{X}_T^U - \widetilde{X}_T\|_2 + \|\widetilde{X}_T - \hat{X}_T^V\|_2. \tag{53}$$

The first pair has identical initial conditions but different inputs. Therefore, from the previous subsection,

$$\|\hat{X}_T^U - \widetilde{X}_T\|_2 \le C_U \sqrt{T\Delta t} \|\Delta U\|_{U,2}. \tag{54}$$

The second pair is driven by the same input V, so $\Delta U = 0$. Equation (34) reduces to

$$\mathcal{V}(t) \le e^{-\alpha t} \mathcal{V}(0). \tag{55}$$

Using Eq. (35),

$$\|\Delta Z(t)\|_2 \le \sqrt{\frac{c_2}{c_1}} e^{-\alpha t/2} \|\Delta Z_0\|_2. \tag{56}$$

Consequently,

$$\begin{aligned} \int_0^{\mathcal{T}} \|\Delta Z(t)\|_2 \, dt &\le \sqrt{\frac{c_2}{c_1}} \|\Delta Z_0\|_2 \int_0^{\infty} e^{-\alpha t/2} \, dt \\ &= \frac{2}{\alpha} \sqrt{\frac{c_2}{c_1}} \|\Delta Z_0\|_2. \end{aligned} \tag{57}$$

The corresponding position contribution is therefore bounded independently of the forecast horizon:

$$\|\Delta P_{\mathcal{T}}\|_2 \le \|\Delta P_0\|_2 + \frac{2L_g}{\alpha}\sqrt{\frac{c_2}{c_1}}\|\Delta Z_0\|_2. \tag{58}$$

Thus there exists a finite constant $C_{X,0}$, independent of T, such that

$$\|\widetilde{X}_T - \hat{X}_T^V\|_2 \le C_{X,0}\|\Delta X_0\|_2. \tag{59}$$

Combining Eqs. (53), (54), and (59) gives

$$\boxed{\|\hat{X}_T^U - \hat{X}_T^V\|_2 \le C_{X,0}\|\Delta X_0\|_2 + C_X\sqrt{T\Delta t}\|\Delta U\|_{U,2}}. \tag{60}$$

For trajectories initialized from the same observed state, $\Delta X_0 = 0$, and

$$\boxed{\|\hat{X}_T^U - \hat{X}_T^V\|_2 \le C_X\sqrt{T\Delta t}\|\Delta U\|_{U,2}}. \tag{61}$$

Therefore, for a fixed sampling interval and perturbations whose input-energy norm remains bounded independently of the horizon, the propagation-induced terminal sensitivity grows at most as $\mathcal{O}(\sqrt{T})$.

The square-root factor arises from accumulating a bounded tracking response through the horizontal position integrator. It does not imply that nominal trajectory-prediction error itself must scale as $\mathcal{O}(\sqrt{T})$, because both future-representation error and its induced perturbation energy may depend on the forecast horizon.

B Implementation and Experimental Details

B.1 Dataset Representation and Evaluation Protocol

The raw ADS-B trajectories are based on the TrajAir dataset (Patrikar et al., 2022), which contains real aircraft observations from non-towered terminal airspace. TrajAir was collected over 111 days spanning approximately eight months and includes ADS-B trajectory information together with contextual aviation data.

We represent each aircraft state as

$$X_t = [x_t, y_t, h_t, v_t, \psi_t, v_{z,t}]^\top, \tag{62}$$

where (x_t, y_t) is horizontal position, h_t altitude, v_t ground speed, ψ_t heading, and $v_{z,t}$ vertical speed.

The prepared data are sampled at 1 Hz. Each forecasting example contains

$$H = 40 \tag{63}$$

historical states and

$$T = 120 \tag{64}$$

future states. The resulting training, validation, and test sets contain 78,308, 25,650, and 21,816 forecasting windows, respectively.

A single 120 s prediction is produced for each test example. The 30, 60, 90, and 120 s metrics are computed from prefixes of the same prediction rather than by independently decoding a new trajectory for every horizon.

All methods are evaluated on the same ordered test windows. Neural-model results are reported over three independent training seeds. The reported sample standard deviation describes optimization and initialization variability rather than a confidence interval over physical flights.

Table 3: Principal RPL representation architecture settings.

Configuration	Value
Sampling interval	1 s
History length	40 steps
Maximum evaluated horizon	120 steps
Encoder input dimension	12
Encoder hidden dimension	256
Latent dimension	64
Encoder layers	4
Encoder attention heads	8
Reference-decoder hidden dimension	256
Reference-decoder layers	2
Reference-decoder attention heads	8
Dropout	0.1
Maximum supported model horizon	300 steps

B.2 RPL Architecture

Table 3 summarizes the principal architecture settings used by the representation module.

The 12-dimensional encoder input combines local trajectory-motion information with normalized absolute spatial context. The additional spatial-context variables include absolute horizontal position, altitude, and sine/cosine heading features. Multi-second geometric estimates are used for motion-related quantities to reduce sensitivity to one-second ADS-B differentiation noise.

The history encoder produces the parameters of the latent distribution

$$q_\theta(z \mid H_t), \tag{65}$$

with $z \in \mathbb{R}^{64}$. During deterministic evaluation, the corresponding deterministic latent estimate is used.

B.3 Kinematically Consistent Reference Generation

The reference decoder predicts a geometric trajectory rather than six independent physical-state channels. Specifically, it predicts

$$\left(x_k^{\mathrm{ref}}, y_k^{\mathrm{ref}}, h_k^{\mathrm{ref}}\right), \tag{66}$$

while speed, heading, and vertical speed are derived differentiably from the geometry.

For a finite-difference window of w samples,

$$v_k^{\mathrm{ref}} = \frac{\left\| P_k^{\mathrm{ref}} - P_{k-w}^{\mathrm{ref}} \right\|_2}{w\Delta t}, \tag{67}$$

$$\psi_k^{\mathrm{ref}} = \mathrm{atan2}\left(y_k^{\mathrm{ref}} - y_{k-w}^{\mathrm{ref}}, x_k^{\mathrm{ref}} - x_{k-w}^{\mathrm{ref}}\right), \tag{68}$$

and

$$v_{z,k}^{\mathrm{ref}} = \frac{h_k^{\mathrm{ref}} - h_{k-w}^{\mathrm{ref}}}{w\Delta t}. \tag{69}$$

The implementation uses a 5 s geometry window for deriving the reference kinematics. Recent observed geometry is also used to provide the forecast-start motion anchor.

The geometric reference is represented as a bounded residual around the history-derived flight-motion baseline. The decoder bounds its residual to

$$30\ \mathrm{km} \tag{70}$$

in the along-track direction,

$$30\ \mathrm{km} \tag{71}$$

in the cross-track direction, and

$$5\ \mathrm{km} \tag{72}$$

in altitude.

This construction ensures that the reference position and its associated speed, heading, and vertical-rate variables remain internally kinematically consistent.

B.4 Reference Smoothing and Feed-Forward Controls

Before physical propagation, the geometric reference is smoothed to reduce high-frequency variations that would otherwise induce large reference-control derivatives.

The feed-forward control vector is

$$u_k^{\mathrm{ff}} = \begin{bmatrix} a_k^{\mathrm{ff}} \\ \omega_k^{\mathrm{ff}} \\ a_{z,k}^{\mathrm{ff}} \end{bmatrix}, \tag{73}$$

where the three components correspond to longitudinal acceleration, yaw rate, and vertical acceleration.

They are obtained from consecutive reference states as

$$a_k^{\mathrm{ff}} = \frac{v_{k+1}^{\mathrm{ref}} - v_k^{\mathrm{ref}}}{\Delta t}, \tag{74}$$

$$\omega_k^{\mathrm{ff}} = \frac{\mathrm{wrap}\left(\psi_{k+1}^{\mathrm{ref}} - \psi_k^{\mathrm{ref}}\right)}{\Delta t}, \tag{75}$$

and

$$a_{z,k}^{\mathrm{ff}} = \frac{v_{z,k+1}^{\mathrm{ref}} - v_{z,k}^{\mathrm{ref}}}{\Delta t}. \tag{76}$$

The control bounds used by the propagation module are

$$|a_k| \le 3.0\ \mathrm{m/s^2}, \tag{77}$$

$$|\omega_k| \le 0.05\ \mathrm{rad/s}, \tag{78}$$

and

$$|a_{z,k}| \le 1.0\ \mathrm{m/s^2}. \tag{79}$$

Reference-derived commands outside these ranges are projected onto the admissible control set $\mathcal{U}$.

B.5 Closed-Loop Physical Propagation

The tracking state is

$$Z_k = (h_k, v_k, \psi_k, v_{z,k}), \tag{80}$$

and the tracking error is

$$e_k = Z_k^{\mathrm{ref}} - \hat{Z}_k, \tag{81}$$

with the heading component computed using wrapped angular difference.

The bounded control law used in the main formulation is

$$u_k = \Pi_{\mathcal{U}}\left(u_k^{\mathrm{ff}} + g_k \odot \pi_K(e_k)\right), \qquad g_k \in [0,1]^{d_u}. \tag{82}$$

The adaptive gate controls how strongly the feedback correction is applied. In the corresponding no-gate ablation, this modulation is removed while retaining the remaining realization pipeline.

The resulting command is propagated through the differentiable reduced-order aircraft dynamics. Writing

$$u_k = (a_k, \omega_k, a_{z,k})^\top, \tag{83}$$

the associated continuous-time kinematic field is

$$f(X_k, u_k) = \begin{bmatrix} v_k \cos\psi_k \\ v_k \sin\psi_k \\ v_{z,k} \\ a_k \\ \omega_k \\ a_{z,k} \end{bmatrix}. \tag{84}$$

The sampled deterministic propagator can therefore be written compactly as

$$\hat{X}_{k+1} = F_{\Delta t}(\hat{X}_k, u_k), \qquad \hat{X}_0 = X_t. \tag{85}$$

The physical module is deliberately reduced order. Its purpose is not to reproduce a full six-degree-of-freedom aerodynamic simulator, but to expose the kinematic structure that couples speed, heading, vertical motion, and accumulated position. Related data-driven aircraft-dynamics models based on Neural ODEs have also demonstrated the utility of combining physical structure with learned dynamics (Jarry et al., 2025).

B.6 Training Details

RPL is trained in stages so that future representation and physical realization can be evaluated separately.

Reference stage. The representation encoder and reference generator are optimized before the physical-realization stage. The representation configuration uses AdamW with learning rate

$$10^{-4}, \tag{86}$$

weight decay 10^{-2}, batch size 128, gradient clipping at 1, and early stopping on validation performance.

The reference loss primarily supervises horizontal position, altitude, and terminal horizontal position. Because speed, heading, and vertical rate are derived from the predicted geometric curve, they are not treated as independent reference-output channels.

Realization stage. The representation source is then used to train the physical realization module. This stage optimizes final-trajectory accuracy together with reference realization, control-effort, and smoothness objectives. The general objective can be written as

$$\mathcal{L} = \lambda_{\text{traj}}\mathcal{L}_{\text{traj}} + \lambda_{\text{ref}}\mathcal{L}_{\text{ref}} + \lambda_{\text{latent}}\mathcal{L}_{\text{latent}} + \lambda_{\text{stab}}\mathcal{L}_{\text{stab}} + \lambda_{\text{ctrl}}\mathcal{L}_{\text{ctrl}}, \tag{87}$$

where different terms are activated or weighted according to the training stage.

The component ablations in Section 4.3 use the same representation source so that changes in performance can be attributed to the realization components rather than to a different learned future representation.

B.7 Controlled Disturbance Protocol

The robustness experiments evaluate the propagation mechanism under controlled perturbations of the tracking variables

$$Z = (h, v, \psi, v_z). \tag{88}$$

The implementation uses nominal initial perturbation scales

$$\sigma_{\text{init}} = [5.0,\ 0.8,\ 0.01,\ 0.15], \tag{89}$$

and per-step process perturbation scales

$$\sigma_{\text{proc}} = [0.15,\ 0.05,\ 5 \times 10^{-4},\ 0.02], \tag{90}$$

for altitude, speed, heading, and vertical rate, respectively. Gaussian draws are clipped at three standard deviations.

A common severity multiplier scales the perturbation magnitude. Open- and closed-loop rollouts receive identical disturbance realizations for each paired sample. This paired construction isolates the contribution of feedback rather than comparing trajectories driven by different random draws.

The disturbance experiments are controlled stress tests rather than calibrated models of real atmospheric uncertainty.

B.8 Theory-Aligned Perturbation Diagnostics

For fixed-reference input perturbations, the normalized tracking gain is

$$G_{\mathrm{trk}}(H) = \frac{\left(\sum_{t=1}^{K_H} \|\delta Z_t\|_2^2\right)^{1/2}}{\left(\sum_{t=1}^{K_H} \|\delta U_t\|_2^2\right)^{1/2}}. \tag{91}$$

The terminal horizontal-position gain is

$$G_{\mathrm{pos}}(H) = \frac{\|\delta P_{K_H}\|_2}{\left(\sum_{t=1}^{K_H} \|\delta U_t\|_2^2\right)^{1/2}}. \tag{92}$$

To directly examine the square-root accumulation mechanism, we also report

$$\frac{G_{\mathrm{pos}}(H)}{\sqrt{T_H}}. \tag{93}$$

In the corresponding diagnostic figure, solid curves denote the median over perturbation trials and dashed curves denote the 95th percentile. Thus the dashed curves characterize the upper tail of perturbation sensitivity rather than a different model or controller.

The local Jacobian audit uses

$$J_k = \left.\frac{\partial F_{\mathrm{cl}}}{\partial Z}\right|_{(Z_k, U_k)}, \tag{94}$$

with spectral radius

$$\rho(J_k) = \max_j |\lambda_j(J_k)|. \tag{95}$$

These quantities are local diagnostics only. A sampled spectral radius near one or a violated common quadratic metric does not establish or refute a global nonlinear contraction property.

B.9 Baseline Implementation and Relation to Prior Work

The LSTM, Transformer, PatchTST, ASCENT, CVAE, and CFM baselines are trained independently of RPL and do not share the RPL representation network. This separation is important because it ensures that improvements attributed to RPL do not arise from sharing its latent representation or reference generator with the competing methods.

Aircraft trajectory prediction has been studied extensively with recurrent and hybrid neural architectures. LSTM-based aircraft predictors have been investigated in both convolution-enhanced and sequence-prediction settings (Pang et al., 2019; Zeng et al., 2020), while CNN–LSTM architectures provide another representative approach to four-dimensional trajectory prediction (Ma & Tian, 2020). Broader supervised deep-learning studies have compared multiple neural architectures for aircraft forecasting (Schimpf et al., 2023). These works motivate the recurrent baseline family used in our evaluation.

Beyond single-aircraft sequence modeling, interaction-aware formulations have been developed for collaborative multi-aircraft prediction (Xu et al., 2021). TrajAir and its associated socially-aware prediction setting provide a representative benchmark for aircraft interactions in non-towered terminal airspace (Patrikar et al., 2022). Other studies explicitly consider uncertainty or degraded observations. Weather-aware Bayesian prediction incorporates meteorological uncertainty into trajectory forecasting

(Pang et al., 2021), while neural prediction under ADS-B failure conditions examines robustness to incomplete or unreliable surveillance information (Yang et al., 2023).

Transformer-based aircraft prediction has also become increasingly prominent. Patched spatial–temporal Transformer models have been used for trajectory change prediction (Hao et al., 2024), while ASCENT develops a Transformer-based multimodal aircraft predictor for non-towered terminal airspace (Prutsch et al., 2026). PatchTST provides a complementary general time-series forecasting baseline based on patch-level Transformer representations (Nie et al., 2023). Together, these approaches motivate the Transformer and PatchTST baselines used in our experiments.

Generative trajectory forecasting provides a second major line of comparison. Conditional variational autoencoders provide a general framework for multimodal structured prediction (Sohn et al., 2015). In aircraft forecasting specifically, long-horizon trajectory generation has been studied using generative adversarial networks (Wu et al., 2022), followed by generative deep-learning approaches for four-dimensional trajectory prediction (Zhang & Liu, 2024) and extensions that combine generative modeling with clustering (Zhang & Liu, 2025). GooDFlight introduces goal-oriented diffusion modeling for aircraft trajectory prediction (Yang et al., 2025), while conditional flow matching has recently been applied to short-term aircraft trajectory generation (Figuet et al., 2026). Our CVAE and CFM baselines represent this generative family, with the latter adapting the general flow-matching formulation of Lipman et al. (2023) to the aircraft forecasting task.

The aircraft-prediction literature also contains approaches that incorporate explicit physical or operational structure. Earlier work estimated aircraft mass, thrust, and operational factors to improve model-based climb prediction (Alligier et al., 2013; Alligier & Gianazza, 2018). More recently, Neural ODE models have been explored for learning aircraft flight dynamics directly from data (Jarry et al., 2025). These approaches illustrate the value of combining observational data with dynamical structure, although their objectives differ from the representation–propagation decomposition considered here.

More broadly, RPL is related to several developments outside aircraft forecasting. Trajectron++ incorporates dynamic feasibility into trajectory forecasting (Salzmann et al., 2020). Neural ODEs, Latent ODEs, and Neural CDEs provide continuous-time frameworks for learning dynamical evolution (Chen et al., 2018; Rubanova et al., 2019; Kidger et al., 2020), while Hamiltonian neural networks and Symplectic ODE-Net introduce structure-preserving inductive biases for learned dynamics (Greydanus et al., 2019; Zhong et al., 2020). Universal differential equations further combine mechanistic structure with flexible learned components (Rackauckas et al., 2020). RPL shares the general motivation of combining learning and structure but does not primarily aim to identify a complete continuous-time dynamical model.

RPL is additionally connected to differentiable prediction–control interfaces and stability-aware learning. Differentiable MPC and differentiable physics enable gradients to pass through structured planning, control, or physical simulation modules (Amos et al., 2018; Belbute-Peres et al., 2018), while differentiable predictive control learns control policies through differentiable closed-loop models (Drgoňa et al., 2022). Contraction and incremental-stability methods provide tools for analyzing the sensitivity of nonlinear dynamical systems (Angeli, 2002; Lohmiller & Slotine, 1998; Tsukamoto & Chung, 2021a;b; Tsukamoto et al., 2021; Li et al., 2025).

Finally, recent work has drawn increasing attention to long-horizon rollout degradation. Scheduled sampling addresses train–test mismatch in sequential prediction (Bengio et al., 2015); learned-dynamics studies have examined long-horizon prediction for physical systems (Rao et al., 2024), full-horizon rollout in learned dynamics models (Lin et al., 2026), and stability-regularized approaches for robust long-horizon forecasting (Li & Wan, 2026). Related analyses of residual Neural ODE models have also investigated rollout-error propagation and contraction diagnostics (Bachar, 2026).

The distinction in RPL is that these ideas are organized around an explicit representation–propagation interface. The learned representation stage produces a future-motion hypothesis, whereas the propagation stage realizes that hypothesis through bounded closed-loop aircraft dynamics. This decomposition makes it possible to distinguish uncertainty in the future representation from the additional sensitivity introduced during physical rollout, which is the quantity studied theoretically in Section 3 and empirically in the perturbation experiments.