

Article

U-STAR-PIML: Uncertainty-Aware Staged Trust-Adaptive Residual Physics-Informed Machine Learning for Recursive Fixed-Wing Unmanned Aerial Vehicle Dynamics Prediction

Ziran Guo, Zhi Zhu * , Mingxuan Li, Boquan Zhang  and Tao Wang 

College of Systems Engineering, National University of Defense Technology, Changsha 410073, China

* Correspondence: zhuzhi@nudt.edu.cn

Highlights

What are the main findings?

- U-STAR-PIML combines incomplete flight-dynamics knowledge with learned temporal correction and state-dependent one-step predictive uncertainty within a unified unmanned fixed-wing dynamics model.
- Relative to the deliberately compact isolated physics prior, learned temporal correction produces lower one-step, rollout-position, and 20 s prediction errors. An external zero-shot evaluation on 10 independent real-flight logs shows a lower pooled all-state one-step RMSE than Persistence, while fast-state transfer remains mixed. Among the prescribed shifts, wind produces the largest degradation under the simulator-provided known-wind protocol, while 95% one-step coverage ranges from 96.68% to 100%.

What are the implications of the main findings?

- The results demonstrate the value of combining structured physical knowledge with data-driven residual learning when an available flight-dynamics model captures the dominant mechanisms but cannot fully represent operating-condition-dependent discrepancies.
- The proposed approach provides a methodological basis for model-based simulation, state prediction, and prediction-oriented analysis of unmanned fixed-wing systems.

Abstract

This study proposes Uncertainty-Aware Staged Trust-Adaptive Residual Physics-Informed Machine Learning (U-STAR-PIML), a recursive one-step dynamics model for fixed-wing unmanned aerial vehicles, evaluated with the JSBSim C172x as a surrogate simulation benchmark. It combines a learnable compact six-degree-of-freedom prior, history-dependent temporal residual correction, trust-aware state-dependent residual gating, hard kinematics, a heteroscedastic one-step uncertainty head, and staged optimization. The protocol separates one-step accuracy, recursive rollout, predictive-interval behavior, and physically distinct distribution shifts. Across five training seeds, the full U-STAR-PIML model (E5) achieves a mean one-step root-mean-square error (RMSE) of 0.005515 ± 0.000009 and the lowest mean rollout-position RMSE of 18.61 ± 2.15 m; the data-driven baseline has the lowest 20 s all-state RMSE of 2.923 ± 0.623 . Recursive rankings remain seed-sensitive, without a universal winner. For the representative E5 cross-condition evaluation, the exact wind vector used in the JSBSim simulation is supplied to the model at every prediction step, i.e., perfect wind information is assumed. Under this assumption, wind out-of-distribution (OOD) conditions cause the largest degradation, with rollout-position RMSE reaching 72.76 m; wind-estimation error is not evaluated. An external zero-shot evaluation on 10 independent IDF-DS Ranger 2400 real-flight logs reduces pooled one-step all-state RMSE



Academic Editors: Salvatore Rosario Bassolillo, Vito Antonio Nardi and Giuseppe Tricomi

Received: 30 July 2026

Revised: 31 August 2026

Accepted: 8 September 2026

Published: 10 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

from 0.12374 for Persistence to 0.03379, although improvements are not uniform across dynamic state groups. The uncertainty head yields 95% empirical coverage of 96.68–100%, with conservative over-coverage under most conditions. These results support simulation-based prediction and an initial cross-airframe transfer diagnostic but do not establish same-airframe sim-to-real transfer, recursive real-flight stability, or operational validity.

Keywords: fixed-wing UAV; flight dynamics modeling; physics-informed machine learning; residual learning; gated fusion; predictive uncertainty; recursive prediction; distribution shift; JSBSim

1. Introduction

Fixed-wing flight dynamics prediction requires accurate nonlinear state-transition models for simulation, state estimation, control-oriented analysis, and recursive prediction. The transition couples body-axis velocity, angular rate, attitude, and position, allowing errors in one state group to propagate through the others. A useful model must resolve fast dynamic coupling while preserving the slower position evolution produced by repeated integration. Aircraft modeling research therefore increasingly combines aerodynamic knowledge with learned representations [1,2]. At the mission level, recent studies involving multiple unmanned aerial vehicles (UAVs) have improved cooperative reconnaissance and search using belief probability maps and double-critic deep reinforcement learning [3], while objective-programming-based partitions and rule-based spanning-tree coverage have been developed for balanced regional coverage planning [4]. These advances illustrate the growing sophistication of high-level UAV autonomy and reinforce the need for reliable lower-level state-transition models to support simulation and mission execution. This study focuses on flight dynamics state-transition learning rather than higher-level planning or control-policy design. Throughout this study, the target domain is restricted to unmanned fixed-wing aircraft. The JSBSim C172x model is employed solely as a surrogate flight-dynamics environment for generating controlled simulation trajectories; this study does not address piloted-aircraft operation or draw conclusions regarding manned aviation.

Compact six-degree-of-freedom (6DoF) equations preserve interpretable rigid-body structure, but their fidelity depends on aerodynamic, inertial, propulsion, disturbance, and actuator models. Wind, parameter variations, and control delay can alter the effective state transition. Observation noise instead affects the state information supplied to a predictor rather than the underlying simulator dynamics itself. The present study evaluates the former transition- and input-level shifts under a clean-state protocol; robustness to noisy onboard state estimates is outside the reported evaluation. Because dynamics mismatches vary across operating conditions, a single nominal parameterization may not represent all transitions with comparable fidelity. The mismatch can also vary across state groups, leaving position evolution sensitive to errors originating in faster states. Learned temporal models can capture unresolved nonlinear effects from data, yet purely data-driven formulations depend on training-distribution coverage and may not preserve physical consistency [1,5]. When predicted states are fed back recursively, small local discrepancies can compound into long-horizon drift. The modeling problem is therefore to retain useful physical structure while learning condition- and history-dependent corrections that remain effective during recursive use.

Physics-informed machine learning (PIML) and residual hybrid learning provide a natural compromise between incomplete physical models and flexible data-driven prediction [6–8]. Four coupled challenges remain: the physical prior is informative but imperfect; model discrepancy may depend on temporal history; the relative influence of the physical

prior and learned correction may vary across states; and deterministic, recursive, and probabilistic objectives must be optimized together. A flexible residual can compensate for physical error, but it may also obscure when the prior remains informative and complicate probabilistic learning. These challenges intensify when supervision spans quantities with different scales and prediction horizons. Physics-informed training can exhibit gradient imbalance, heterogeneous convergence, and ill-conditioned loss landscapes [9,10]. These difficulties motivate a formulation that represents temporal discrepancy and state-dependent fusion while separating optimization tasks that otherwise compete within a single objective.

Existing flight dynamics studies usually address subsets of deterministic modeling, state or parameter estimation, residual correction, predictive uncertainty, and long-horizon prediction. Reviewing these studies against five capabilities that a recursively deployed dynamics model requires—an explicit physical prior, use of temporal history, free-running recursive prediction, a probabilistic output, and testing under prescribed distribution shift—shows that individual capabilities are well developed but rarely combined, as summarized in Table 1. Three gaps follow from that comparison and define the scope of this work: history-dependent discrepancy correction is seldom paired with state-dependent residual modulation between the physical and learned terms; a probabilistic transition output is seldom evaluated together with free-running recursive rollout; and prescribed shifts of different physical origin are seldom separated within one protocol, so a single aggregate out-of-distribution (OOD) score is reported instead. High one-step accuracy does not guarantee recursive stability, and one-step interval calibration does not establish calibration over an entire predicted trajectory [11,12]. Interval behavior must therefore be tested under distinct shifts that alter the model inputs or underlying dynamics through different mechanisms. The present study evaluates predictive accuracy and one-step interval behavior under three predefined effective shifts—wind, aircraft parameters, and control delay—rather than OOD sample detection. The stored-noise diagnostic subset is retained only as a control because its noisy-observation channels are not consumed by the reported model. A suitable framework must assess local dynamics, recursive behavior, and one-step predictive diagnostics without treating performance in one category as evidence for the others.

To address this coupled problem, this study proposes Uncertainty-Aware Staged Trust-Adaptive Residual Physics-Informed Machine Learning (U-STAR-PIML) as a recursive one-step dynamics learner for fixed-wing UAVs. It combines a learnable compact 6DoF prior with history-dependent temporal residual correction, trust-aware fusion with state-dependent residual gating, and hard attitude and position kinematics. A state-dependent one-step uncertainty head represents heteroscedastic transition variance, while a staged optimization procedure separates deterministic dynamics learning from uncertainty-head training; interval calibration is evaluated post hoc rather than optimized through a separate calibration loss. The formal framework name is retained in this manuscript's title, whereas the operational mechanism is described conservatively as trust-aware gated fusion. The resulting design preserves an interpretable dynamical prior while permitting memory-dependent correction and one-step predictive intervals across varied operating conditions. The model is evaluated under domain-randomized in-distribution (ID) conditions, three effective OOD shifts, and a stored-noise diagnostic using JSBSim C172x episodes. The complete formulation and training procedure are given in Section 3.

First, this study develops a recursively usable physics–data fusion model for fixed-wing dynamics by combining a learnable compact 6DoF prior with history-dependent temporal residual correction. A state-dependent residual gate modulates the learned correction, while attitude and position kinematics are enforced by construction. This

formulation preserves useful physical organization while allowing unresolved dynamics to be learned for recursive one-step state prediction.

Second, a six-stage optimization strategy separates deterministic dynamics learning from one-step predictive-uncertainty learning. The deterministic predictor is first trained using state, derivative, residual, and recursive-rollout supervision, after which it is frozen and the heteroscedastic uncertainty head is optimized with Gaussian negative log-likelihood. This staged design reduces interference between mean prediction and variance estimation while retaining a clear one-step interpretation of the uncertainty output.

Third, this study establishes an evaluation protocol that separately assesses one-step accuracy, recursive rollout, predictive-interval behavior, and physically distinct distribution shifts. Wind, parameter, and control-delay shifts are evaluated together with a diagnostic stored-noise control, while five predefined training seeds quantify run-to-run variability. This protocol distinguishes local accuracy, long-horizon behavior, and predictive uncertainty without reducing them to a single overall model ranking.

These contributions address recursive dynamics prediction in simulation-based fixed-wing UAV applications where the available physical model is informative but incomplete. The resulting metrics provide complementary evidence for selecting prediction models according to horizon and failure mode without implying closed-loop control or flight-safety validation.

The remainder of this manuscript reviews related studies, presents the proposed methodology, reports the case study results, and discusses the principal findings and limitations.

2. Related Work

2.1. Fixed-Wing UAV Dynamics Modeling and Recursive Prediction

Physics-based fixed-wing models preserve force–moment and rigid-body structure, supporting interpretation and parameter estimation, but their accuracy remains tied to aerodynamic model form and parameter fidelity [2,13]. Data-driven temporal models can instead learn nonlinear state transitions from measured histories and have supported aircraft modeling with flight test validation [14]. The two categories therefore trade explicit physical organization against flexibility in representing unmodeled effects. Their recursive use, however, exposes accumulated errors that may be inconspicuous under one-step evaluation, particularly when the learned transition is applied beyond the conditions represented in training.

Existing aircraft and UAV studies span physics-based neural flight modeling, data-driven identification, and physics-informed state prediction or estimation [15,16]. The reported platforms include fixed-wing aircraft and multi-rotor UAVs, with outputs ranging from deterministic states and parameters to probabilistic flight dynamics predictions [17–20]. Temporal history and free-running rollout are not treated uniformly across these formulations. Existing work therefore provides useful deterministic modeling and identification methods, whereas recursive fixed-wing state-transition prediction that jointly addresses model discrepancy and uncertainty has received less systematic study.

2.2. Physics-Informed Machine Learning for Flight Dynamics

Physics-informed neural networks (PINNs) and broader PIML formulations incorporate prior knowledge through soft equation penalties, hard physical relations, hybrid physical–neural architectures, or learnable physical parameters [6,21]. Hard relations guarantee selected constraints by construction, whereas soft penalties leave their satisfaction dependent on optimization. In flight dynamics, these mechanisms primarily support deterministic state prediction, state estimation, and parameter identification, with a smaller

body of work addressing probabilistic outputs [17,18,20]. Because these tasks impose different data and recursion requirements, performance in trajectory reconstruction or inverse estimation provides limited evidence for stable stochastic state-transition prediction.

Combining data and physics losses can produce gradient imbalance, unequal convergence rates, and ill-conditioned optimization landscapes [9,10,22]. In a joint objective, readily fitted terms may dominate before weakly observed physical relations have converged. These difficulties motivate staged or adaptive training when state, derivative, physical, and rollout objectives interact. Under the reported flight dynamics protocols, however, PIML remains oriented mainly toward deterministic reconstruction or identification. History-dependent model error, recursive stability, and probabilistic transition learning are therefore rarely addressed together within one flight dynamics formulation.

2.3. Residual Hybrid Models and Adaptive Fusion

Hybrid dynamics models range from full data-driven replacement to residual correction of incomplete physics and joint physics–data prediction. Residual learning is well suited to cases in which a physical model captures dominant structure but omits persistent effects [8,23,24]. Memoryless corrections use only the current state, whereas discrepancies associated with wind history, unobserved aerodynamic states, actuator lag, or control delay can require recurrent or window-based context. Temporal residuals increase representational capacity, but their value must ultimately be assessed under recursive use.

Direct residual addition applies a fixed interaction between physical and learned terms, while state-dependent gating permits input-dependent residual modulation. Such modulation requires matched ablation because gating need not improve every metric independently of residual capacity or training design. Fourier and general neural operators provide an alternative representation of temporal histories [25,26]; in this manuscript, the operator variants serve as controlled extensions rather than replacements for the primary model. The combined use of temporal residual correction, adaptive fusion, and recursive evaluation remains insufficiently integrated in flight dynamics studies.

2.4. Predictive Uncertainty Under Distribution Shift

Regression uncertainty can be represented through heteroscedastic Gaussian likelihoods, Bayesian or ensemble approximations, and conformal prediction [11,27–29]. Likelihood models provide direct state-dependent scales, Bayesian and ensemble methods approximate parameter uncertainty, and conformal procedures target empirical coverage. Time-series conformal methods further address temporally dependent prediction intervals [30]. These approaches differ in assumptions and computational cost, and an estimated one-step transition variance is not a complete separation of aleatoric and epistemic uncertainty. Reliability is commonly assessed through likelihood, empirical coverage, interval width, and calibration error; coverage and width require joint interpretation because high coverage may reflect conservative intervals.

Calibration under the training distribution does not ensure reliability after distribution shift [12]. Generic OOD research primarily addresses detection of unfamiliar samples [31], whereas dynamics regression evaluates predictive accuracy and interval behavior under predefined physical shifts. Wind, parameter variation, and control delay alter the effective model input or transition dynamics. Observation noise instead constitutes a measurement-level perturbation and is an input-robustness condition only when the predictor consumes the noisy measurements. In the present study, the stored noisy observations are not network inputs and are therefore retained only as a diagnostic control. Moreover, recursive feedback alters future model inputs, so one-step calibration does not establish trajectory-level

uncertainty calibration. Current flight dynamics studies provide limited joint assessment of these shifts with recursive prediction.

2.5. Synthesis and Positioning of This Work

Four gaps emerge from these methodological categories. Flight-dynamics PIML primarily addresses deterministic prediction, estimation, or identification; hybrid residual models do not always represent memory-dependent discrepancy and state-dependent residual modulation jointly; and probabilistic transition modeling is often evaluated separately from recursive rollout. Common prescribed-shift protocols that separate wind, parameter, and delay while jointly examining derivative behavior, recursive drift, and calibration analysis remain limited; observation-noise conditions additionally require explicit confirmation that the predictor consumes the noisy measurements. Table 1 summarizes representative learning-based flight dynamics models against the five capabilities introduced in Section 1.

Table 1. Learning-based flight dynamics models assessed for recursive deployment.

Study	Physics	History	Rollout	Uncert.	Shift
NeuroBEM [15] (quad.)	✓	×	○	×	○
PI-TCN [16] (quad.)	○	✓	○	×	○
Neural-Fly [24] (quad.)	✓	○	×	×	✓
UQ-PINN [20] (fixed-wing)	✓	×	✓	✓	×
PINN est. [18] (quad.)	✓	×	×	×	×
This work (fixed-wing)	✓	✓	✓	✓	✓

Note: Physics, History, Rollout, Uncert., and Shift denote the five assessed capabilities. Symbols indicate explicit support (✓), limited or indirect support (○), or an unaddressed capability (×); “quad.” denotes multi-rotor platforms. Ratings summarize reported scope, not quality. The present model uses clean states, so its stored-noise subset is diagnostic rather than an input-noise robustness test. Boldface indicates the present study.

U-STAR-PIML is positioned at the intersection of physics-guided flight dynamics, temporal residual learning, trust-aware gated fusion, state-dependent one-step predictive uncertainty, and recursive evaluation under ID and prescribed distribution shifts, providing a unified basis for examining these coupled capabilities under a consistent fixed-wing simulation protocol.

3. Methodology

This section presents the proposed U-STAR-PIML framework for fixed-wing UAV dynamics prediction. The method is designed as a recursive one-step dynamics learner rather than a direct time-to-trajectory regressor. It integrates a learnable 6DoF physics branch, a temporal residual dynamics branch, a trust-aware fusion module with state-dependent residual gating, and a heteroscedastic one-step uncertainty head. The methodology is organized into three parts. Section 3.1 introduces the overall framework. Section 3.2 describes the JSBSim-based data generation process and the physics-guided dynamics model. Section 3.3 presents the loss function design and the staged training algorithm [17,18,21].

3.1. Overall Framework of U-STAR-PIML

The overall framework of U-STAR-PIML is shown in Figure 1. The domain-randomized flight records contain clean simulator states, control commands, wind disturbances, noisy observations, observation-availability indicators, actuation delays, and randomized physical parameters. For the reported training and evaluation runs, the configured model input consists of the clean simulator-state, command, and wind sequences; the stored noisy-observation channels and availability indicators are not consumed by the network. The model contains two parallel dynamics branches. The first branch is a

learnable 6DoF physics branch that provides a physically meaningful prior derivative. The second branch is a temporal residual branch that receives a rolling history buffer and learns the mismatch between the simplified physics prior and the JSBSim-generated dynamics. The globally scaled physics derivative is combined additively with the state-gated residual correction. A separate one-step uncertainty head then supplies a state-dependent predictive variance for the fused mean.

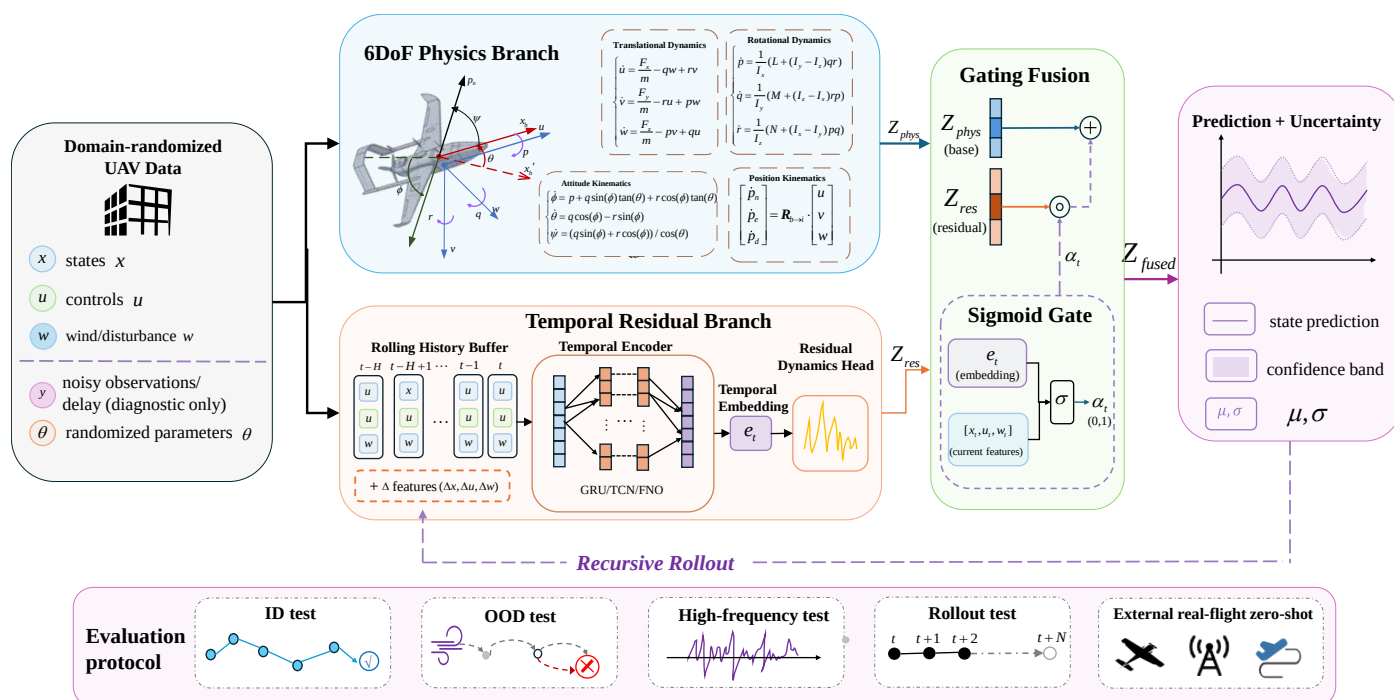


Figure 1. U-STAR-PIML architecture for fixed-wing UAV dynamics prediction. The left block summarizes the generated flight records. Only the clean simulator states, controls, and wind are supplied to the predictor; stored noisy observations are retained only for the diagnostic subset and are not network inputs, while delay and randomized physical parameters modify the generated episodes rather than constituting additional predictor inputs. The E5 model combines globally scaled learnable 6DoF physics with a state-dependent residual-gated GRU-based temporal correction to predict the one-step mean and state-dependent one-step variance. E6–E8 use TCN/FNO encoders. Positions use the NED frame.

Let $\mathbf{x}_t \in \mathbb{R}^{12}$ denote the clean JSBSim aircraft state, $\mathbf{u}_t \in \mathbb{R}^4$ denote the control input, and $\mathbf{w}_t \in \mathbb{R}^3$ denote the wind disturbance. Given a history length H , the tensor interface used by the model at time t is represented as

$$\mathcal{I}_t = \{\mathbf{x}_{t-H+1:t}, \mathbf{u}_{t-H+1:t}, \mathbf{w}_{t-H+1:t}, \mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t\}. \quad (1)$$

where the current triplet $(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t)$ is the final (H -th) element of the corresponding history and is also passed separately to the physics branch, current-feature heads, and gate; that is, the length- H history window ends at and includes the current time step t , and the current triplet is listed again not as an additional independent observation but to indicate this separate routing. The history sequences therefore have dimensions $\mathbf{x}_{t-H+1:t} \in \mathbb{R}^{H \times 12}$, $\mathbf{u}_{t-H+1:t} \in \mathbb{R}^{H \times 4}$, and $\mathbf{w}_{t-H+1:t} \in \mathbb{R}^{H \times 3}$. In one-step training and evaluation, every window is constructed from clean simulator-state records and the supervised target is the clean next state $\mathbf{x}_{t+\Delta t}$; this is a clean-state transition protocol rather than an observation-to-state estimation protocol. The model predicts the next-state mean and the corresponding

state derivative. Conditioned on the input information $\mathcal{I}_t$, the uncertainty output is defined by the standard conditional Gaussian model

$$\mathbf{x}_{t+\Delta t} \mid \mathcal{I}_t \sim \mathcal{N}(\boldsymbol{\mu}_{t+\Delta t}, \boldsymbol{\Sigma}_t), \quad (2)$$

where $\boldsymbol{\mu}_{t+\Delta t}$ is the one-step predicted mean and $\boldsymbol{\Sigma}_t$ is its diagonal covariance. The uncertainty head directly outputs a state-dependent log-variance vector $\mathbf{s}_t \in \mathbb{R}^{12}$:

$$\boldsymbol{\Sigma}_t = \text{diag}(\exp(\mathbf{s}_t)), \quad \mathbf{s}_t = h_\sigma(\mathbf{e}_t, \mathbf{c}_t), \quad (3)$$

where $\mathbf{e}_t$ is the temporal embedding and $\mathbf{c}_t$ denotes the current feature vector formed by $\mathbf{x}_t$, $\mathbf{u}_t$, and $\mathbf{w}_t$. The uncertainty head therefore defines a diagonal heteroscedastic Gaussian model for the one-step state transition. Unlike denoising diffusion probabilistic models or score-based generative models [32,33], it does not construct an iterative noising–denoising chain or solve a reverse-time stochastic differential equation. It directly outputs a conditional variance for a single state transition, and neither samples nor covariance are propagated during recursive rollout. The output is therefore interpreted as state-dependent one-step predictive uncertainty, not as a generative diffusion process or a stochastic trajectory model.

For engineering interpretation, $\boldsymbol{\Sigma}_t$ provides an auxiliary indication of the local dispersion associated with the next-state prediction under the current model and input conditions. It can attach state-wise intervals to one-step estimates and support downstream prediction monitoring or conservative decision logic. This study does not validate the output as an automatic fallback trigger, safety controller, collision-risk threshold, certification bound, or other formal safety guarantee.

The deterministic part of the model is defined by a fused drift. Since the physics and residual branches mainly predict the six dynamic states, the state vector is divided into dynamic and kinematic components:

$$\mathbf{x} = [\mathbf{x}_{\text{dyn}}^T, \mathbf{x}_{\text{kin}}^T]^T, \quad \mathbf{x}_{\text{dyn}} = [u, v, w, p, q, r]^T, \quad \mathbf{x}_{\text{kin}} = [\phi, \theta, \psi, p_n, p_e, p_d]^T. \quad (4)$$

The fused dynamic derivative is computed as

$$\dot{\mathbf{x}}_{t,\text{dyn}}^{\text{fused}} = \tau \dot{\mathbf{x}}_{t,\text{dyn}}^{\text{phys}} + \boldsymbol{\alpha}_t \odot \dot{\mathbf{x}}_{t,\text{dyn}}^{\text{res}}, \quad (5)$$

where $\dot{\mathbf{x}}_{t,\text{dyn}}^{\text{phys}}$ is the physics-induced dynamic derivative, $\dot{\mathbf{x}}_{t,\text{dyn}}^{\text{res}}$ is the learned residual dynamic derivative, τ is the global learnable physics scaling coefficient, $\boldsymbol{\alpha}_t \in (0, 1)^6$ is the state-dependent residual gate, and $\odot$ denotes element-wise multiplication. The term “trust-aware” does not denote a normalized or explicit online redistribution of reliability between the physics and learned branches. Instead, the learned scalar τ controls the global contribution of the physical prior, whereas the state-dependent gate $\boldsymbol{\alpha}_t$ regulates the magnitude of the residual correction according to the temporal and current flight features. Accordingly, τ is interpreted as a learned global physics-scaling coefficient and $\boldsymbol{\alpha}_t$ as a state-dependent residual-gating coefficient, rather than as complementary probabilities or normalized trust scores. The two coefficients are not constrained to sum to one. Each component of $\boldsymbol{\alpha}_t$ is produced by a sigmoid and then floored at a small positive value (0.02 in the reported configuration) so that $\boldsymbol{\alpha}_t \in [0.02, 1)^6$ and the residual branch is never fully switched off; the upper bound remains below one because the sigmoid saturates there. The scalar τ is shared across the six dynamic states, parameterized through a logit so that

$\tau = \sigma(\cdot) \in (0, 1)$, initialized at 0.35, and updated jointly with the network. The kinematic derivative is enforced by hard kinematic relations:

$$\dot{\mathbf{x}}_{t,\text{kin}}^{\text{fused}} = h_{\text{kin}}(\mathbf{x}_t). \quad (6)$$

Therefore, the full fused derivative is

$$\dot{\mathbf{x}}_t^{\text{fused}} = \left[\left(\dot{\mathbf{x}}_{t,\text{dyn}}^{\text{fused}} \right)^T, \left(\dot{\mathbf{x}}_{t,\text{kin}}^{\text{fused}} \right)^T \right]^T. \quad (7)$$

The one-step mean prediction is obtained by numerical integration:

$$\boldsymbol{\mu}_{t+\Delta t} = \mathbf{x}_t + \Delta t \dot{\mathbf{x}}_t^{\text{fused}}. \quad (8)$$

During recursive evaluation, a clean simulator-state history of length H provides the warm start. After the rollout boundary, the predicted state mean $\hat{\mathbf{x}}_{t+\Delta t} = \boldsymbol{\mu}_{t+\Delta t}$ is fed back into the rolling state-history buffer; no sample from the one-step uncertainty output is fed back. The control and wind histories continue to follow the recorded reference sequences. Future clean simulator states are retained only as evaluation targets.

Equation (8) uses an explicit forward-Euler update, which is the default integrator for the present sampling interval; an optional fourth-order Runge–Kutta variant that reuses the same control, wind, and history inputs over the step is also implemented. At the sampling interval $\Delta t = 0.02$ s used here, the two integrators produce practically indistinguishable one-step updates, so forward Euler is retained for efficiency. The Euler angles are kept in their principal ranges, and the yaw angle ψ is wrapped to $(-\pi, \pi]$ after each update through $\text{atan2}(\sin \psi, \cos \psi)$; all attitude errors used in the loss functions and in the reference-derivative estimation are likewise evaluated on the wrapped angular difference so that discontinuities at $\pm\pi$ do not corrupt the training signal or the recursive state. It should nevertheless be noted that the hard kinematic relation in Equation (6) only guarantees that the instantaneous kinematic derivative is consistent at each step; it does not by itself bound the accumulated integration error over a long recursive horizon, which is instead controlled empirically by the rollout and position-consistency supervision introduced in Section 3.3 and quantified in the horizon-wise analysis of Section 4.3.

3.2. Data Generation and Physics-Guided Dynamics Model

The flight trajectories are generated using JSBSim, an open-source flight dynamics model widely used for aircraft simulation and flight-control research [34]. The proposed U-STAR-PIML framework is developed exclusively for unmanned fixed-wing aircraft. The JSBSim C172x model is adopted solely as a reproducible surrogate six-degree-of-freedom flight-dynamics environment for developing and evaluating the proposed UAV prediction method. Although the underlying C172x configuration represents a conventional light-aircraft model, it is not treated as the target application platform. All problem definitions, prediction tasks, evaluation protocols, and intended applications in this study are formulated for unmanned aircraft, and no conclusions are drawn regarding piloted-aircraft operation.

The state vector is $\mathbf{x} = [u, v, w, p, q, r, \phi, \theta, \psi, p_n, p_e, p_d]^T$, where the first six entries are body-axis translational velocities and angular rates, the next three are Euler angles, and the final three are north-east-down (NED) positions. The control vector is $\mathbf{u} = [\delta_e, \delta_a, \delta_r, \delta_t]^T$ for elevator, aileron, rudder, and throttle commands, and the wind vector is $\mathbf{w} = [w_n, w_e, w_d]^T$ in the NED frame. These definitions are given inline because they specify the data interface rather than an additional model assumption.

To construct a dynamics learning dataset spanning varied simulated operating conditions, domain randomization is applied during JSBSim simulation [35]. Each flight episode independently samples wind parameters, control-excitation profiles, initial conditions, observation noise, actuation delay, and selected physical-parameter scales. The generator includes ID presets for training, validation, and testing, as well as dataset presets for stronger wind disturbance, parameter perturbation, increased stored observation noise, and control delay. Under the reported clean-state configuration, the wind, parameter, and delay presets alter the model inputs or underlying dynamics, whereas the increased-noise preset changes only the stored observation channels and is therefore used as a diagnostic subset. All episodes are split at the episode level to avoid leakage between adjacent time samples.

For the ID subsets, the randomized factors remain within the nominal training range. The three effective OOD subsets enlarge one factor at a time: wind disturbance, physical-parameter scales, or command delay. The stored-noise diagnostic changes only the stored observation channels. The dataset schema is “episode_id, t , split, $\mathbf{u}_t$, $\mathbf{w}_t$, obs_mask, $\mathbf{x}_t$, $\mathbf{y}_t$ ”. Here, $\mathbf{x}_t$ is the clean JSBSim state and $\mathbf{y}_t$ is a stored noisy observation; the scalar mask indicates whether the complete observation is present. The reported loader selects $\mathbf{x}_t$ with use_obs_mask = false, so the stored-noise diagnostic changes neither the network input nor the clean target.

The raw horizontal-position channels produced by the generator are stored relative to each episode’s starting point, so every trajectory begins at the coordinate origin; this introduces an artificial zero boundary at the initial time and an episode-dependent offset that are not part of the aircraft dynamics and would otherwise bias a position-based error metric toward small values near the start of each episode. To remove this artifact, the signed NED position is reconstructed using the kinematic relation

$$\dot{\mathbf{p}}_n = \mathbf{R}_b^n(\phi, \theta, \psi) \mathbf{v}_b, \quad \mathbf{p}_n = \begin{bmatrix} p_n & p_e & p_d \end{bmatrix}^T, \quad \mathbf{v}_b = \begin{bmatrix} u & v & w \end{bmatrix}^T. \quad (9)$$

In the implementation, p_n and p_e are reconstructed by trapezoidal integration of the body-to-NED velocity, which yields a signed and physically consistent horizontal track while leaving the body-axis velocities, angular rates, and attitude angles unchanged. This reconstruction affects only the two horizontal-position channels, and it is applied identically to the reference targets and to every model variant; it therefore cannot bias the cross-model comparison. In particular, the one-step grouped RMSE, derivative, high-maneuver, and ablation results are computed from the dynamic and attitude states and are independent of the horizontal-position reconstruction, and the rollout-position RMSE is evaluated for all variants on the same reconstructed coordinates, so their relative ranking is unaffected. The sensitivity of the rollout-position metric to this reconstruction is independently checked against the direct JSBSim local-position reference in Section 4.3. The corresponding noisy-observation columns are reconstructed while preserving their original observation-noise offsets, although those columns are not selected by the reported model configuration.

The physics branch follows a compact 6DoF fixed-wing rigid-body formulation. Standard translational, rotational, attitude, and position relations are retained, with body-frame force $\mathbf{F}_b$, moment $\mathbf{M}_b$, mass m , inertia matrix $\mathbf{J}$, and gravity $\mathbf{g}_b$ carrying their usual flight-dynamics meanings. Their detailed forms are provided in Appendix A, while the compact representation used by the proposed method is

$$\dot{\mathbf{x}}_t^{\text{phys}} = f_{6\text{DoF}}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t; \boldsymbol{\theta}_p, \boldsymbol{\lambda}_p), \quad (10)$$

where $\boldsymbol{\theta}_p$ denotes nominal physical parameters and $\boldsymbol{\lambda}_p$ denotes learnable physics scaling coefficients. These coefficients are global across episodes, so episode-to-episode variation is absorbed by the temporal residual branch rather than by a separate per-episode physics

model. This design keeps the model close to the 6DoF prior while allowing adaptation to simulator-specific dynamics and randomized parameters.

The temporal residual branch learns the discrepancy between the physics-induced dynamic derivative and the data-implied dynamic derivative. The implementation first constructs the normalized per-step feature vector

$$\mathbf{z}_k = \left[S_x(\mathbf{x}_k)^T, S_u(\mathbf{u}_k)^T, S_w(\mathbf{w}_k)^T \right]^T, \quad \mathbf{e}_t = E_\psi(\mathbf{z}_{t-H+1:t}, \Delta \mathbf{z}_{t-H+1:t}), \quad (11)$$

where S_x , S_u , and S_w are fitted from the training episodes only. The first history difference is set to zero, and subsequent differences are computed inline as $\Delta \mathbf{z}_k = \mathbf{z}_k - \mathbf{z}_{k-1}$ after normalization. The residual dynamic derivative is then obtained as

$$\dot{\mathbf{x}}_{t,\text{dyn}}^{\text{res}} = R_\psi(\mathbf{e}_t). \quad (12)$$

The state-dependent residual gate is computed from the temporal embedding and current features:

$$\alpha_t = \sigma[G_\omega(\mathbf{e}_t, \mathbf{c}_t)], \quad \mathbf{c}_t = \left[S_x(\mathbf{x}_t)^T, S_u(\mathbf{u}_t)^T, S_w(\mathbf{w}_t)^T \right]^T, \quad (13)$$

where $\sigma(\cdot)$ is the sigmoid function. By construction, the gate α_t depends on the current normalized state, control, and wind features in $\mathbf{c}_t$ and on the temporal embedding $\mathbf{e}_t$. This makes state- and history-dependent residual modulation available to the model but does not prescribe a monotonic response to wind or an explicit online estimate of branch reliability. Values of α_t near one indicate that the residual correction is passed with little attenuation, whereas lower values indicate stronger attenuation of the residual correction; they do not directly represent higher or lower physics trust. In the main U-STAR-PIML model, E_ψ is implemented as a recurrent temporal encoder. In the temporal-operator extension, the encoder can be replaced by a Fourier neural operator to study spectral temporal encoding, motivated by neural-operator studies that learn mappings between function spaces and parameterize integral kernels in Fourier space [25,26]. This extension is used only as a controlled methodological variant and does not replace the main staged U-STAR-PIML model.

3.3. Training Process and Algorithm Design

The training objective is designed to balance one-step state prediction, derivative consistency, fast-state dynamics, high-frequency increment consistency, residual alignment, recursive rollout stability, navigation-position supervision, gate smoothness and entropy regularization, and uncertainty-head likelihood training, together with a kinematic consistency term that the hard kinematics already satisfy by construction. The total loss is defined as

$$\begin{aligned} \mathcal{L} = & \lambda_{\text{state}} \mathcal{L}_{\text{state}} + \lambda_{\text{der}} \mathcal{L}_{\text{der}} + \lambda_{\text{fast}} \mathcal{L}_{\text{fast}} + \lambda_{\text{slope}} \mathcal{L}_{\text{slope}} + \lambda_{\text{res}} \mathcal{L}_{\text{res}} + \lambda_{\text{kin}} \mathcal{L}_{\text{kin}} \\ & + \lambda_{\text{roll}} \mathcal{L}_{\text{roll}} + \lambda_{\text{freq}} \mathcal{L}_{\text{freq}} + \lambda_{\text{nav}} \mathcal{L}_{\text{nav}} + \lambda_{\text{pos}} \mathcal{L}_{\text{pos}} + \lambda_{\text{gate}} \mathcal{L}_{\text{gate}} + \lambda_{\text{nll}} \mathcal{L}_{\text{nll}} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}}. \end{aligned} \quad (14)$$

Here, $\mathcal{L}_{\text{pos}}$ compactly denotes the position-slope, turning-point, and position-band constraints used to reduce horizontal position drift during recursive rollout. Table 2 summarizes the thirteen terms in Equation (14). Of these, eleven are active data-driven losses; the remaining two, $\mathcal{L}_{\text{kin}}$ and $\mathcal{L}_{\text{nav}}$, are consistency terms that the hard kinematic relation in Equation (6) already satisfies by construction, so they are effectively zero and contribute negligible gradient during training (they are retained as explicit consistency checks rather than as drivers of optimization, as discussed below). For each term, the table lists the

quantity it penalizes, its defining equation, and its weight in each of the six training stages shown in Figure 2; because Stages 3 and 4 share one weight profile and Stages 5 and 6 share another, the per-stage weights reduce to four distinct columns. The principal loss terms are described below, while the detailed auxiliary definitions are provided in Appendix A.2.

Table 2. Stage-wise loss weights in the reported six-stage schedule.

Term	Quantity Penalized	Equation	S1	S2	S3–S4	S5–S6
$\mathcal{L}_{\text{state}}$	Normalized one-step next-state error	(A5)	1.00	1.00	1.00	0.20
$\mathcal{L}_{\text{der}}$	Fused vs. reference state-derivative error	(A6)	0.80	0.80	0.80	0.10
$\mathcal{L}_{\text{fast}}$	Derivative error on fast states $\{v, w, p, q, r\}$	(A7)	1.00	1.00	1.00	0.10
$\mathcal{L}_{\text{slope}}$	One-step increment error on $\{v, p, q, r\}$	(A8)	0.35	0.40	0.40	–
$\mathcal{L}_{\text{res}}$	Residual vs. (reference – physics) derivative	(15)	–	0.08	0.08	–
$\mathcal{L}_{\text{kin}}$	Fused vs. hard kinematic derivative [†]	(A9)	–	0.10	0.10	–
$\mathcal{L}_{\text{roll}}$	Discounted recursive-rollout state error	(17)	0.08	0.10	0.10	–
$\mathcal{L}_{\text{freq}}$	High-frequency one-step increment error	(A11)	0.08	0.08	0.08	–
$\mathcal{L}_{\text{nav}}$	Fused vs. reference position derivative [†]	(A10)	0.12	0.15	0.15	0.05
$\mathcal{L}_{\text{pos}}$	Position slope/turning-point/band constraints [‡]	(A12)	0.08	0.10	0.10	–
$\mathcal{L}_{\text{gate}}$	Temporal smoothness of the gate α_t	(text)	–	–	0.010	–
$\mathcal{L}_{\text{nll}}$	Diagonal Gaussian negative log-likelihood (NLL) of the one-step uncertainty head	(18)	–	–	–	1.00
$\mathcal{L}_{\text{reg}}$	Gate entropy/anti-saturation regularizer	(text)	–	–	0.002	–

Note: S1–S6 denote the stages in Figure 2; S3–S4 and S5–S6 share weight columns. A dash denotes zero, and “text” denotes a prose-defined term. [†] The hard kinematic relation makes $\mathcal{L}_{\text{kin}} \equiv 0$ and $\mathcal{L}_{\text{nav}} \approx 0$. [‡] In S3–S4, $\mathcal{L}_{\text{pos}}$ includes turning-point and band weights of 0.10 and 0.06. In S5–S6, only $\mathcal{L}_{\text{nll}}$ updates the one-step uncertainty head.

The auxiliary deterministic terms are defined from normalized next-state errors, fused-derivative errors, fast-state derivatives on $\mathcal{F} = \{v, w, p, q, r\}$, and one-step slopes on $\mathcal{H} = \{v, p, q, r\}$. The fast-state and slope terms emphasize rapidly changing translational and rotational channels, while the frequency and position terms described below target transient response and accumulated horizontal drift. Their detailed sample-wise expressions are given in Appendix A.2, and Table 2 records the corresponding stage weights. The residual alignment loss constrains the residual branch to learn the discrepancy between the true dynamic derivative and the physics-induced dynamic derivative:

$$\mathcal{L}_{\text{res}} = \left\| S_{\dot{x}, \text{dyn}} \left(\dot{\mathbf{x}}_{t, \text{dyn}}^{\text{res}} \right) - S_{\dot{x}, \text{dyn}} \left(\dot{\mathbf{x}}_{t, \text{dyn}} - \dot{\mathbf{x}}_{t, \text{dyn}}^{\text{phys}} \right) \right\|_2^2. \quad (15)$$

The target of Equation (15) is the plain difference $\dot{\mathbf{x}}_{t, \text{dyn}} - \dot{\mathbf{x}}_{t, \text{dyn}}^{\text{phys}}$, with the physics derivative treated as a fixed reference (its gradient is detached during this stage), which implicitly assumes unit fusion coefficients. This differs from the deployed fusion in Equation (5), where the physics branch is scaled by τ and the residual by α_t . Equation (15) is therefore a directional alignment regularizer rather than a constraint that presupposes $\tau = 1$ and $\alpha_t = 1$. The numerical consistency of the fused derivative with the reference is enforced by the state and derivative losses in Appendix A.2, which act on the fused output and remain active during joint fine-tuning. Because the corresponding attitude and position relations are enforced by construction through Equation (6), $\mathcal{L}_{\text{kin}}$ is identically zero and $\mathcal{L}_{\text{nav}}$ is close to zero in the reported clean-state data. They act as consistency checks and contribute negligible optimization gradients rather than imposing additional soft physical constraints; their explicit forms are provided in Appendix A.2.

To reduce free-running error accumulation, recursive rollout supervision is introduced. The initial rolling buffer is populated with a clean simulator-state history. At

each free-running step, the model uses the current recorded control and wind values and recursively predicts

$$\hat{\mathbf{x}}_{t+k+\Delta t} = F_{\Theta}(\hat{\mathbf{x}}_{t+k-H+1:t+k}, \mathbf{u}_{t+k-H+1:t+k}, \mathbf{w}_{t+k-H+1:t+k}), \quad (16)$$

where $\hat{\mathbf{x}}_{t+k+\Delta t} = \boldsymbol{\mu}_{t+k+\Delta t}$ is the deterministic predicted mean. This mean, rather than a sample from the one-step Gaussian output or a future clean state, is appended to the state-history buffer. Recorded control and wind sequences are appended to their respective history buffers. The rollout loss is

$$\mathcal{L}_{\text{roll}} = \frac{1}{K} \sum_{k=1}^K \gamma^k \|S_x(\hat{\mathbf{x}}_{t+k}) - S_x(\mathbf{x}_{t+k})\|_2^2, \quad (17)$$

where γ is a discount coefficient used to balance near-term and longer-term rollout errors.

The remaining auxiliary terms are implementation-level regularizers. $\mathcal{L}_{\text{freq}}$ emphasizes normalized high-frequency increments, while $\mathcal{L}_{\text{pos}}$ combines maneuver-weighted position-slope, turning-point, and multi-resolution position-band penalties to suppress horizontal drift during recursive rollout. $\mathcal{L}_{\text{gate}}$ smooths adjacent gate values, and $\mathcal{L}_{\text{reg}}$ discourages gate saturation; neither term imposes a convex branch mixture. Detailed definitions and fixed coefficients are collected in Appendix A.2.

The one-step uncertainty output is trained using the diagonal Gaussian NLL [36,37]:

$$\mathcal{L}_{\text{nll}} = \frac{1}{2 \cdot 12} \sum_{j=1}^{12} \left[\frac{(x_{t+\Delta t,j} - \mu_{t+\Delta t,j})^2}{\exp(s_{t,j})} + s_{t,j} \right], \quad (18)$$

where $e_{t,j} = x_{t+\Delta t,j} - \mu_{t+\Delta t,j}$ is the prediction error for state j and $\sigma_{t,j}^2 = \exp(s_{t,j})$ is its predicted variance; the scalar error $e_{t,j}$ is distinct from the temporal embedding $\mathbf{e}_t$. The implementation averages the displayed terms over states and samples. With the deterministic mean predictor frozen during Stages 5–6, the NLL trains the uncertainty head to match its predicted variance to the residual magnitude of the fixed one-step prediction. The normalized squared-error term penalizes underestimated uncertainty for large residuals, whereas the log-variance term discourages uninformative variance inflation. The NLL is not interpreted as a flight-risk score. In implementation, the log-variance is clipped within a conservative numerical range to prevent unstable heteroscedastic Gaussian likelihood training [29].

Under the diagonal Gaussian assumption, the two-sided interval for state j is $\mu_j \pm 1.645\sigma_j$ at 90% and $\mu_j \pm 1.96\sigma_j$ at 95%. Each state has its own σ_j , so each interval is interpreted in that state's physical coordinate and unit. For example, the units are m/s for u , rad/s for q , rad for θ , and m for position. Aggregate NLL and MPIW combine heterogeneous state coordinates and are therefore mixed-coordinate evaluator diagnostics rather than a single physically dimensioned uncertainty magnitude.

The full U-STAR-PIML model is trained through a staged optimization strategy rather than direct end-to-end training. The reason is that deterministic dynamics fitting, physics-residual alignment, gate learning, recursive stability, and uncertainty-head learning have different optimization characteristics. The staged training process is summarized in Figure 2.

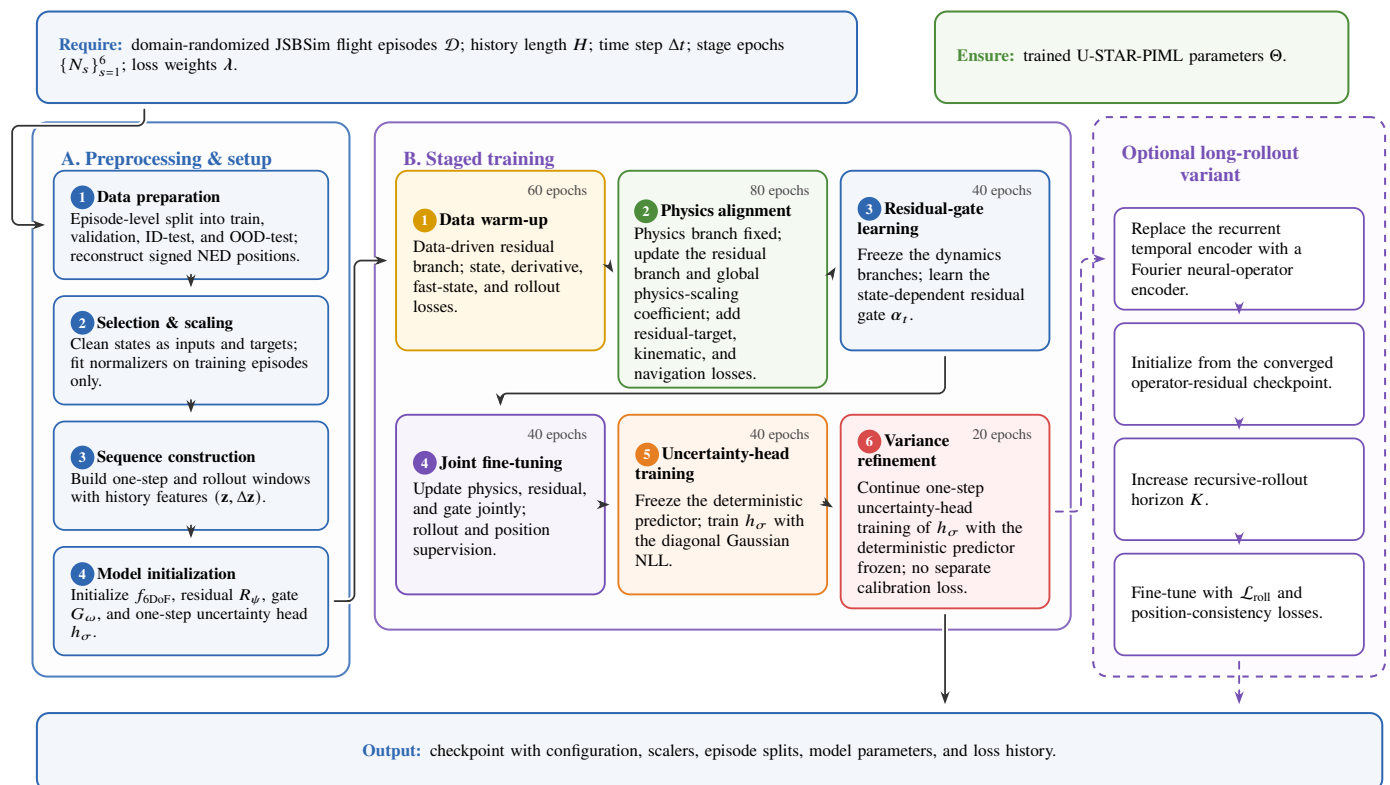


Figure 2. U-STAR-PIML training procedure. Panel A summarizes dataset splitting, training-only normalization, window construction, and module initialization. Panel B trains deterministic drift in Stages 1–4 and continues one-step uncertainty-head training in Stages 5–6 with the predictor frozen. Table 2 lists the weights; the dashed operator branch applies only to E6–E8.

The first four stages establish a stable deterministic drift predictor. The fifth and sixth stages then train the state-dependent transition variance under a diagonal Gaussian likelihood; in the reported configuration, the sixth stage continues the same uncertainty-only objective rather than applying a separate calibration loss. Stage 6 is therefore an extended uncertainty-head training phase, and the resulting interval calibration is assessed only post hoc in Section 4.4. Throughout all six stages, the model is fitted on the training episodes and the validation episodes are used only for monitoring and checkpoint selection; the ID-test and OOD-test episodes are never used for fitting, for checkpoint selection, or for any adjustment of the one-step uncertainty output, so the reported interval statistics are obtained on data that the model has not seen in any form. This staged design prevents the uncertainty head from compensating for an under-trained mean predictor and associates its output with one-step residual transition variability under the clean-state protocol. The optional temporal-operator long-rollout variant is treated as an extension for studying long-horizon recursive stability, while the main method remains the staged U-STAR-PIML model.

4. Case Study

4.1. Experimental Setup

This section evaluates the proposed U-STAR-PIML framework for unmanned fixed-wing aircraft using the JSBSim C172x model solely as a surrogate flight-dynamics simulation benchmark. The evaluation focuses on model behavior under ID conditions, three effective prescribed shifts, one stored-noise diagnostic control, recursive prediction, dynamic consistency, one-step predictive uncertainty, and ablation-based component analysis. The section is organized around the experimental setup, data source, evaluation protocol, and

model comparison design, thereby providing a reproducible validation protocol for the methodology described in Section 3.

Flight trajectories are generated with the JSBSim C172x surrogate model, which provides a reproducible six-degree-of-freedom simulation environment with a consistent interface and controlled randomization conditions. The data are produced with JSBSim version 1.2.4, installed through the Python 3.10 package `jsbsim`, using the bundled `c172x` aircraft definition. No aircraft-model files are modified; mass, inertia, aerodynamic, and air-density variations are applied through JSBSim properties at run time. The generated trajectories are used exclusively to evaluate unmanned fixed-wing aircraft dynamics prediction; they are not intended to reproduce piloted-aircraft missions or human-in-the-loop operation.

Each flight episode is generated independently and stores the clean state sequence, control inputs, wind disturbance information, noisy observations, and a scalar observation-availability indicator. The reported E5 training and evaluation configuration uses the clean state, control, and wind sequences for one-step and recursive prediction. The observation-related fields remain available for dataset diagnostics and future state-estimation protocols. This simulation setting provides a controlled environment for evaluating unmanned fixed-wing dynamics prediction while preserving the principal rigid-body flight dynamics structure used in recent fixed-wing UAV modeling studies [13,34,38].

Data generation uses a sampling interval of $\Delta t = 0.02$ s, corresponding to a 50 Hz integration rate, and each formal simulation episode lasts 40 s. The model input history length is $H = 50$, corresponding to 1.0 s of historical information, and the one-step prediction interval is 0.02 s. The main training configuration uses a batch size of 256 and a random seed of 42. The same seed drives the per-episode data-generation randomization, making the episode split and all sampled parameters reproducible.

Each episode is initialized near a level-flight trim condition at an altitude of approximately 1 km. The initial state is drawn around this trim by adding independent Gaussian perturbations to the body-axis velocity (standard deviation 0.5 m s^{-1} on u , clipped to $[42, 58] \text{ m s}^{-1}$, and 0.02 m s^{-1} on v and w), the angular rates ($0.0015 \text{ rad s}^{-1}$), the attitude angles (0.004 rad on ϕ and θ , with θ clipped to $[-0.15, 0.15] \text{ rad}$, and 0.010 rad on ψ), and the down position (1 m). The control commands are normalized flight-control-system inputs in $[-1, 1]$: a trim throttle sampled in $[0.54, 0.63]$ is combined with small sinusoidal excitation, and one excitation profile is drawn per episode from an elevator doublet, an aileron doublet, a rudder doublet, a throttle step, and a multi-axis chirp, with fixed moderate amplitudes chosen to keep the C172x response stable while preserving excitation diversity. The wind disturbance follows a three-axis Ornstein–Uhlenbeck process whose per-episode mean and standard deviation are sampled from the ranges in Table 3 and whose correlation time is drawn from $[3, 10]$ s. Because these settings, together with the JSBSim version and aircraft model, fully determine the generated data, the complete per-subset randomization ranges are listed in Table 3, and the remaining fixed generator constants are available from the corresponding author upon reasonable request.

External real-flight validation. The external evaluation uses the public IDF-DS fixed-wing UAV telemetry dataset [39]. We selected the Volantex Ranger 2400 subset recorded with a Holybro Pixhawk 6X running PX4 and evaluated the frozen E5 model without adaptation. A performance-independent deterministic protocol selected 10 independent source logs, yielding 29,500 valid one-step samples at 50 Hz, with a 50-sample history and a 0.02 s prediction interval. Normalized controls were reconstructed from PX4 `actuator_servos` and `actuator_motors` records; `actuator_outputs` pulse-width modulation (PWM) records were excluded. Horizontal wind used the onboard PX4 `airspeed_wind` estimate, and the unavailable vertical component was set to zero. No real-flight data were used for training,

fine-tuning, or test-time adaptation, and recursive real-flight rollout was not evaluated. One candidate log was excluded before model evaluation because its actuator-channel behavior was incompatible with the fixed control-surface mapping used by the validation protocol; model performance was not used for this decision. Because the Ranger 2400 differs from the simulated C172x, this is a combined simulator-to-real and cross-airframe transfer diagnostic rather than same-airframe operational validation.

The dataset comprises ID subsets, three effective OOD subsets (wind, parameter, and delay), and one stored-noise diagnostic subset, with 98 formal flight episodes in total. The ID data include 40 training episodes, 8 validation episodes, and 10 ID test episodes. The ID test episodes follow the same randomization range as the training episodes and are reserved exclusively for evaluation. The three effective OOD subsets are generated by enlarging one model-input or transition-dynamics factor while keeping the remaining factors close to the nominal randomization range. Specifically, `ood_wind` contains 10 episodes and expands the wind-process range from the ID wind mean interval $[-0.10, 0.10]$ m/s and wind standard-deviation interval $[0.05, 0.60]$ m/s to the wind mean interval $[-2.0, 2.0]$ m/s and wind standard-deviation interval $[1.00, 2.50]$ m/s. The `ood_param` subset contains 10 episodes with mass and inertia scale factors in $[0.82, 1.18]$, aerodynamic scale factors in $[0.85, 1.15]$, and air-density scale factors in $[0.92, 1.08]$. The dataset internally named `ood_noise` contains 10 episodes and increases the observation-noise multiplier to 3.0 in the stored observation channels only. Because these channels are not consumed by the reported model, this subset is referred to throughout this manuscript as the stored-noise diagnostic rather than as an effective OOD robustness test. The `ood_delay` subset contains 10 episodes and uses control-input delays of 3–12 simulation steps, corresponding to 0.06–0.24 s. All subsets are split at the episode level to prevent leakage from temporally adjacent samples. The dataset configuration is summarized in Table 3.

Table 3. Case-study datasets.

Dataset	Episodes	Total Duration (s)	Total Time Steps	Main Randomization Range
ID train	40	1600	80,040	Nominal randomization; wind mean $[-0.10, 0.10]$ m s ⁻¹ , wind std $[0.05, 0.60]$ m s ⁻¹
ID validation	8	320	16,008	Nominal randomization (same range as ID train)
ID test	10	400	20,010	Nominal randomization (same range as ID train)
OOD wind	10	400	20,010	Wind mean $[-2.0, 2.0]$ m s ⁻¹ , wind std $[1.00, 2.50]$ m s ⁻¹
OOD parameter	10	400	20,010	Mass/inertia scale $[0.82, 1.18]$, aerodynamic scale $[0.85, 1.15]$, air-density scale $[0.92, 1.08]$
Stored-noise diagnostic	10	400	20,010	Observation-noise multiplier 3.0 in stored observations only; model inputs remain clean
OOD delay	10	400	20,010	Control-input delay 3–12 steps (0.06–0.24 s; nominal otherwise)

Note: All subsets use the C172x surrogate simulation model and episode-level splits. The generated trajectories are used exclusively for evaluating unmanned fixed-wing aircraft dynamics prediction. Each 40 s episode sampled at $\Delta t = 0.02$ s contains 2001 stored time steps. ID subsets share nominal randomization ranges. Parameter-OOD factors are dimensionless multipliers; nontargeted factors remain nominal.

Two complementary prediction modes are considered. In one-step prediction, each example uses a clean simulator-state history $\mathbf{x}_{t-H+1:t}$ together with the aligned control and wind histories, and the target is the clean next state $\mathbf{x}_{t+\Delta t}$. Each example is independently rebuilt from the recorded sequence, so its history is unaffected by previous model

predictions. This oracle-state protocol assumes that a deployment system would provide an equivalent state estimate; replacing clean simulator states with noisy or intermittent onboard estimates is outside the present evaluation.

In recursive rollout, the first $H = 50$ clean states provide a 1.0 s warm start. Thereafter, the deterministic predicted mean is appended to the state-history buffer, while controls and winds follow the recorded reference sequence. Future clean states are used only to compute errors, and the one-step uncertainty output is not sampled during rollout.

The model receives the exact wind vector used in the JSBSim simulation at every prediction step in both protocols, corresponding to perfect knowledge of the simulated wind, rather than an estimated or forecast wind field. This idealization excludes wind-estimation error because an airborne system would have to infer local wind from onboard measurements. The results should therefore be interpreted as an upper bound with respect to wind information. Evaluating wind inputs that are perturbed, delayed, or withheld is left for future work.

The baseline recursive training configuration uses 8 rollout steps. The standard evaluation horizons are 0.2 s, 1.0 s, 3.0 s, and 5.0 s, with extended horizons of 10.0 s and 20.0 s reported in the long-horizon analyses. The operator-based long-horizon variant uses 50 rollout training steps. The operator rollout fine-tuning variant uses 250 rollout steps, corresponding to 5.0 s of recursive supervision. The 10 epochs reported for E8 denote additional rollout fine tuning after initialization from the converged 45-epoch E7 checkpoint, rather than the complete E8 training budget.

Because $\mathbf{y}$ and obs_mask are not selected by the evaluated configuration, the stored-noise diagnostic multiplier changes only the stored noisy-observation columns. It does not perturb the network inputs, clean targets, or reported clean-state metrics. Accordingly, the stored-noise diagnostic condition evaluates a dataset-storage control rather than robustness to noisy state inputs. The one-step protocol emphasizes local clean-state transition accuracy, whereas the recursive protocol evaluates deterministic error accumulation and free-rollout stability over longer horizons.

The evaluation metrics are selected to cover state accuracy, dynamic consistency, long-horizon stability, maneuver-sensitive behavior, one-step interval behavior, and computational cost. State prediction is evaluated using the root-mean-square error (RMSE) and grouped RMSE over physically related state variables. Derivative consistency is evaluated using derivative RMSE. Recursive prediction is evaluated using horizon-wise RMSE. High-maneuver or fast-state response is assessed using high-maneuver RMSE or fast-state RMSE. The one-step uncertainty output is evaluated using negative log-likelihood (NLL), prediction interval coverage probability (PICP), mean prediction interval width (MPIW), and calibration error. Computational efficiency is characterized by the parameter count, one-step inference time, and recursive rollout time per step. Numerical values are reported only in the corresponding result subsections.

The primary experiment is denoted as E5: Full U-STAR-PIML, which corresponds to the complete staged U-STAR-PIML framework. This model uses a six-stage training schedule with 60, 80, 40, 40, 40, and 20 epochs, respectively, resulting in 280 training epochs in total. The comparison and ablation experiments are named sequentially from E1 to E8 to compare outcomes associated with the physics branch, temporal residual branch, state-dependent residual gating, one-step uncertainty output, staged training, and operator-based temporal extensions. Because several configurations differ simultaneously in architecture, optimization schedule, and total training budget, these experiments constitute partially coupled comparisons rather than strictly matched ablations. The compared experiments and their training schedules are summarized in Table 4.

Table 4. Sequential comparison and ablation experiments.

Experiment	Main Component	Epochs	Purpose
E1: Physics-only baseline	Physics branch only	60	Evaluate the standalone 6DoF prior
E2: Data-driven baseline	Data-driven temporal branch	80	Evaluate data-driven temporal learning
E3: Physics-residual model	Physics plus residual learning	100	Evaluate residual compensation over the physics prior
E4: State-dependent gated fusion model	End-to-end gated fusion	100	Evaluate state-dependent residual gating
E5: Full U-STAR-PIML	Full U-STAR-PIML	280	Evaluate the proposed staged model and one-step uncertainty head
E6: Operator data model	Operator-based data branch	45	Evaluate operator-only temporal encoding
E7: Operator residual model	Operator-based residual branch	45	Evaluate operator residual learning
E8: Operator rollout model	Operator rollout fine-tuning	45 + 10	Evaluate long-horizon rollout fine-tuning

Note: E8 starts from the 45-epoch E7 checkpoint and adds 10 rollout-fine-tuning epochs, yielding 55 cumulative epochs.

Run-to-run reproducibility was assessed using five predefined training seeds (42, 142, 242, 342, and 442), while the episode-level training, validation, ID-test, and OOD partitions were held fixed using split seed 42. E2–E8 were independently trained and evaluated for each seed under their original method-specific optimization schedules, yielding 35 complete runs. For each metric, the analysis reports the sample mean, sample standard deviation (SD; $n = 5$, denominator $n - 1$), and two-sided 95% Student's t confidence interval with four degrees of freedom. Because the method-specific epoch schedules were retained, this repeated-run analysis evaluates sensitivity to training stochasticity rather than providing a strictly matched-budget architecture-only comparison. The experiments therefore remain partially coupled comparisons rather than strictly matched ablations.

4.2. Overall Prediction Performance Across Evaluation Conditions

This subsection reports the overall prediction performance of the complete U-STAR-PIML model, denoted as E5: Full U-STAR-PIML in Table 4. The evaluation compares the ID test subset with three effective model-input or dynamics shifts—wind, parameter, and delay—and one stored-noise diagnostic subset. The purpose of this subsection is to provide an aggregate performance overview across these test conditions while preserving the implementation boundary of the stored-noise diagnostic subset. Detailed analyses of dynamic consistency, recursive error accumulation, one-step interval behavior, and ablation behavior are reserved for Sections 4.3–4.5.

The ID test subset follows the same randomization range as the training data, while its episodes are excluded from model training. The wind-OOD subset strengthens the wind disturbance process. The parameter-OOD subset enlarges perturbations in mass, inertia, aerodynamic coefficients, and air density. The delay-OOD subset introduces larger control-input delays. The stored-noise diagnostic subset contains an increased observation-noise level only in the stored observation columns, which are not consumed by the reported clean-state model. Consequently, wind, parameter, and delay form the effective stress conditions, whereas the stored-noise diagnostic subset is retained as a diagnostic control.

The ID test case serves as the nominal generalization reference for the complete model. Under this condition, one-step prediction assesses whether the learned state-transition map remains accurate on unseen episodes sampled from the training distribution, while recursive rollout evaluates whether the same local model can be used repeatedly without immediate loss of trajectory-level coherence. The ID row in Table 5 therefore provides the reference level for interpreting the three effective OOD shifts and the separate diagnostic control. A representative in-distribution prediction result is shown in Figure 3, which is used to visualize the nominal prediction behavior of the full model.

Table 5. Prediction performance of U-STAR-PIML across the evaluated conditions.

Dataset	Body Vel. RMSE	Angular Rate RMSE	Attitude RMSE	Position RMSE	Rollout Pos. RMSE
ID test	0.00321	0.000295	0.000143	0.01054	18.82
OOD wind	0.02853	0.001920	0.000146	0.01072	72.76
OOD parameter	0.00306	0.000266	0.000139	0.01017	15.31
Stored-noise diagnostic	0.00309	0.000267	0.000145	0.01149	12.97
OOD delay	0.00295	0.000348	0.000143	0.01103	13.73

Note: The first four columns are one-step grouped RMSE values; Rollout Pos. RMSE is recursive. Units for body velocity, angular rate, attitude, and position are m s^{-1} , rad s^{-1} , rad, and m, respectively.

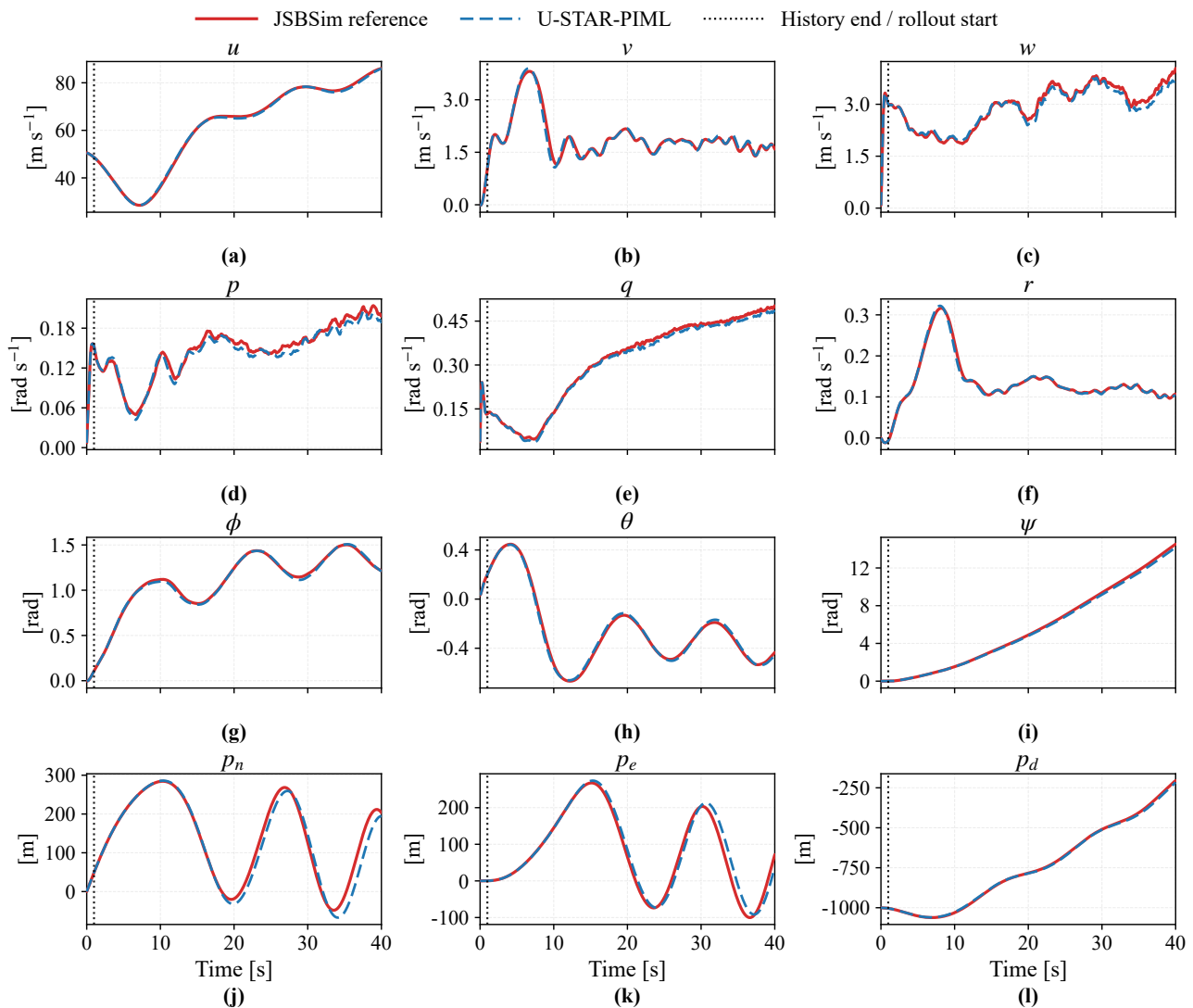


Figure 3. U-STAR-PIML prediction on ID-test episode 48, chosen by fixed evaluator order rather than performance. The dotted line at $t = 0.99$ s separates the 50-step history from rollout starting at 1.00 s. Yaw is unwrapped only for display; model updates and errors use angles wrapped to $(-\pi, \pi]$. Panels (a–c) show the three body-axis velocities, panels (d–f) the three angular rates, panels (g–i) the three Euler angles, and panels (j–l) the three NED positions, with the state order matching the labels shown in each panel.

The representative ID result indicates that the complete model preserves the dominant temporal evolution of the flight states under nominal randomization. This behavior is consistent with the role of the ID test set as a reference case for evaluating generalization to unseen episodes within the training distribution. Since this subsection focuses on overall

prediction performance, detailed derivative-level and long-horizon error-propagation analyses are deferred to Section 4.3.

The effective OOD rows in Table 5 compare the model response to three prescribed shifts. The wind case directly changes the relative airflow experienced by the aircraft and can therefore affect velocity, attitude, and position errors through the aerodynamic response chain. The parameter case evaluates shifted physical properties and aerodynamic scales, while the delay case tests temporally shifted control effects. The stored-noise diagnostic subset contains stronger observation noise only in columns that are not consumed by the reported clean-state configuration. It therefore does not constitute an input-noise shift for the trained network and is retained only as a diagnostic control. Detailed derivative behavior, horizon-wise recursive trends, uncertainty calibration, and component-level ablations are analyzed in later subsections. A representative wind-OD prediction result is shown in Figure 4, providing a visual comparison with the nominal ID behavior.

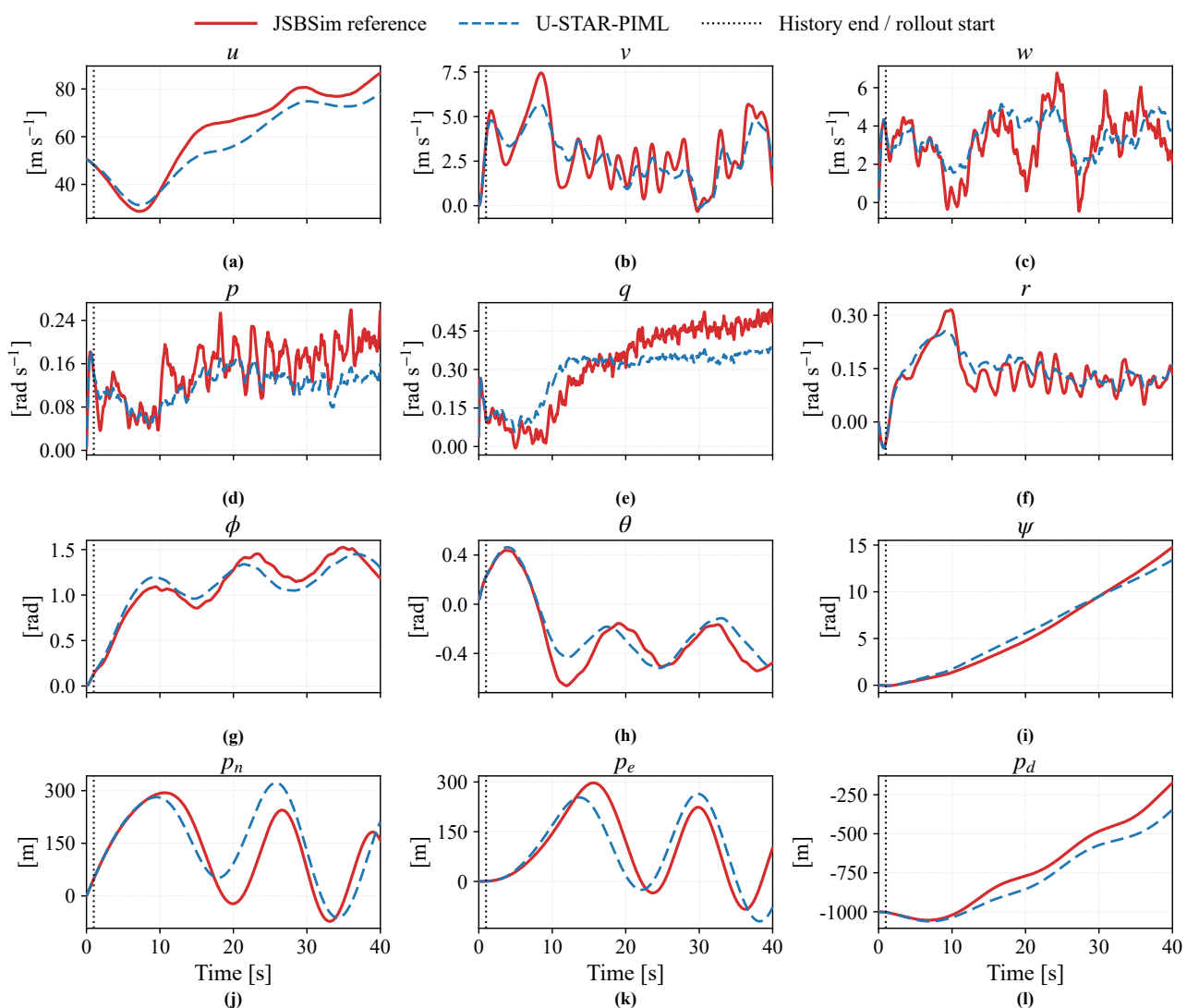


Figure 4. U-STAR-PIML prediction on wind-OD episode 58, chosen by fixed evaluator order rather than performance. The dotted line at $t = 0.99$ s separates the 50-step history from rollout starting at 1.00 s. Yaw is unwrapped only for display; model updates and errors use angles wrapped to $(-\pi, \pi]$.

The wind-OD representative result shows a more challenging prediction condition than the ID case. The strengthened wind process mainly affects the aerodynamic response and therefore tends to increase discrepancies in velocity- and angular-rate-related states.

This visual comparison supports the need for a separate dynamic-consistency and recursive-rollout analysis in the following subsection.

The aggregate cross-case comparison is further summarized in Figure 5.

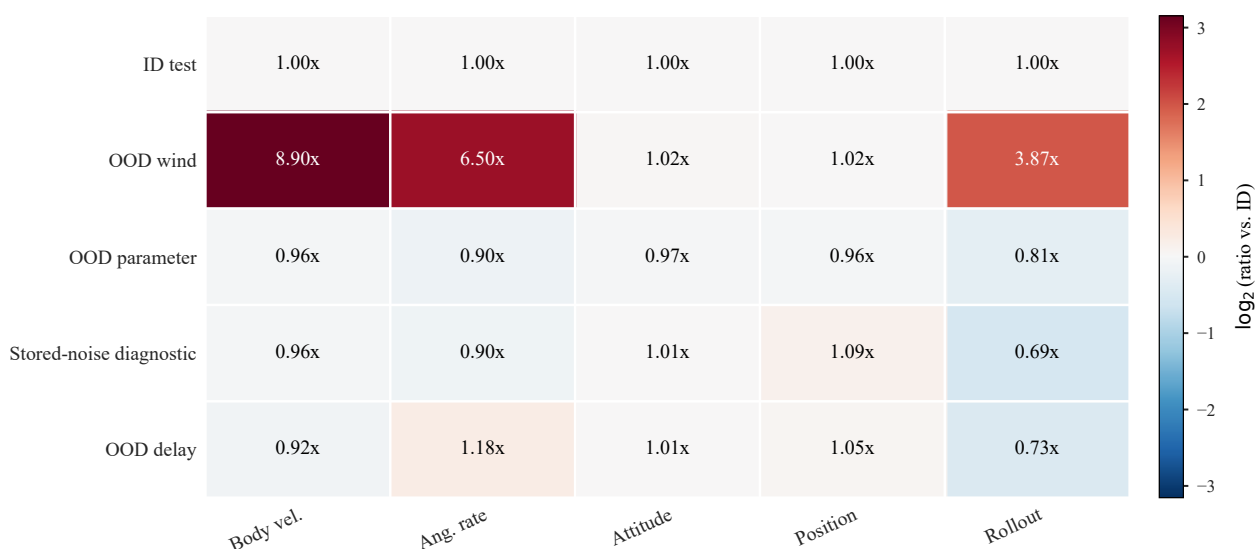


Figure 5. One-step grouped RMSE and recursive position RMSE across the evaluated ID, prescribed-shift, and diagnostic conditions. Cells report dimensionless ratios relative to ID, with ID normalized to 1.00× and color encoding $\log_2(\text{ratio})$. Table 5 gives the absolute values and physical units.

Figure 5 shows that wind produces the largest degradation among the three effective prescribed shifts under the present known-wind protocol. Relative to ID, wind OOD increases body-velocity, angular-rate, and rollout-position RMSE by approximately 8.90, 6.50, and 3.87 times, respectively, with rollout-position RMSE reaching 72.76 m. The pattern is consistent with a disturbance entering through relative airflow, first perturbing velocity and angular-rate prediction, then propagating through attitude kinematics and recursive position integration. For model-based simulation or a prediction module queried repeatedly, this result matters because accurate local transitions can still accumulate material navigation-position drift under persistent aerodynamic disturbance. The implication is bounded to the present known-wind simulation protocol: it does not establish robustness to real-world wind-estimation error or flight conditions. Parameter- and delay-OOD values remain closer to ID, while the stored-noise diagnostic changes only unused observation columns and therefore verifies the data path rather than sensor-noise robustness.

Although the aggregate results identify wind as the most challenging effective prescribed shift, the overall metrics and representative state plots alone do not explain how local dynamic errors propagate into long-horizon trajectory drift. Therefore, the next subsection examines derivative consistency, high-maneuver response, and recursive rollout behavior in greater detail.

4.3. Dynamic Consistency and Recursive Rollout Analysis

Although Section 4.2 provides an aggregate comparison across the evaluated conditions, grouped RMSE values alone do not reveal how local dynamic errors propagate during recursive prediction. This subsection therefore evaluates dynamic consistency and rollout stability using derivative comparison, high-maneuver response, three-dimensional trajectory prediction, and horizon-wise rollout error.

The derivative comparison in Figure 6 uses wind-OOD episode 58, the same fixed-order representative episode shown in Figure 4, to assess local dynamic consistency at

the level of state rates rather than state values alone. This evaluation indicates whether the learned transition model captures the local trend of flight dynamics before errors are accumulated through recursive prediction.

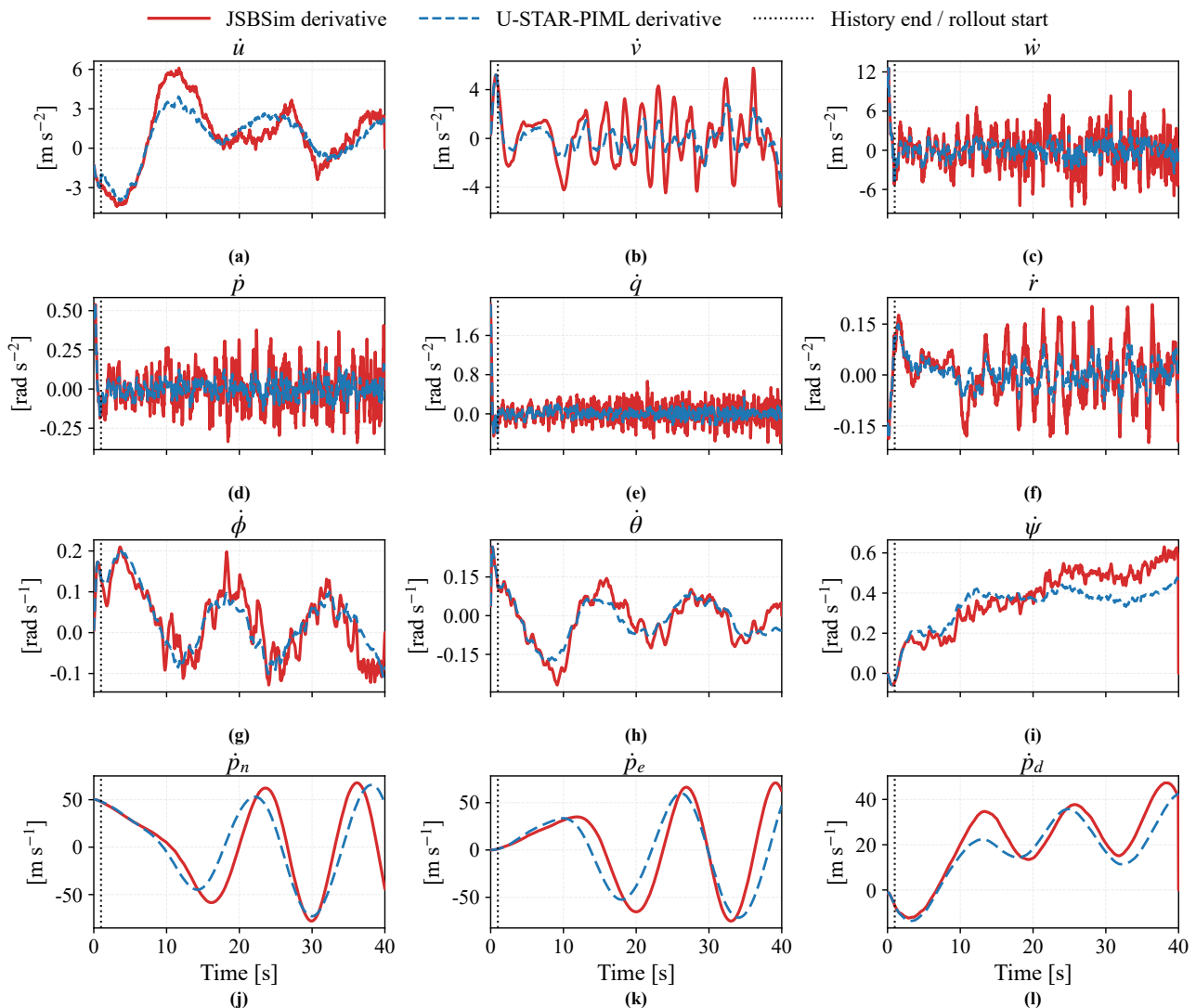


Figure 6. Predicted and finite-difference reference state derivatives for fixed-order wind-OOD episode 58. Reference derivatives use clean JSBSim states. The dotted line at $t = 0.99$ s separates the 50-step history from rollout starting at 1.00 s.

Derivative comparison focuses on the predicted rate of change and therefore complements the state-level prediction results reported in the previous subsection. The reference derivative is not read directly from the simulator: it is estimated from the recorded clean state sequence by finite differences, using a second-order central difference at interior samples and a first-order one-sided difference at the episode endpoints, with the yaw angle ψ unwrapped before differentiation so that the $\pm\pi$ branch cut does not create spurious spikes. Because the differencing is applied to the clean simulator states rather than to noisy observations, and because the sampling interval $\Delta t = 0.02$ s is short relative to the dominant rigid-body time scales, the amplification of measurement noise that normally accompanies numerical differentiation is limited here; the residual discretization error nevertheless affects both the reference and the supervised targets identically, and it should be kept in mind when interpreting the derivative-level agreement. Under wind-OOD conditions, derivatives associated with velocity and angular-rate states are generally ex-

pected to be more sensitive because wind disturbances directly affect relative airflow and aerodynamic response. The purpose of this analysis is to characterize whether the model preserves the local dynamic tendency without expanding the discussion to each individual state component.

High-maneuver recursive behavior was evaluated in all 10 wind-OOD episodes (episodes 58–67). Within each episode, post-warm-start samples at or above the maneuver-score 75th percentile were identified independently. Qualifying intervals separated by no more than 0.10 s were merged, and merged intervals lasting at least 0.20 s were retained. The procedure yielded 157 segments, 3915 samples, and 75.16 s in total. Figure 7 shows the highest-mean-score segment, from episode 63 at 38.22–38.78 s (0.56 s, 29 samples, mean score 0.979). This segment is illustrative; the dataset-level assessment uses all retained segments.

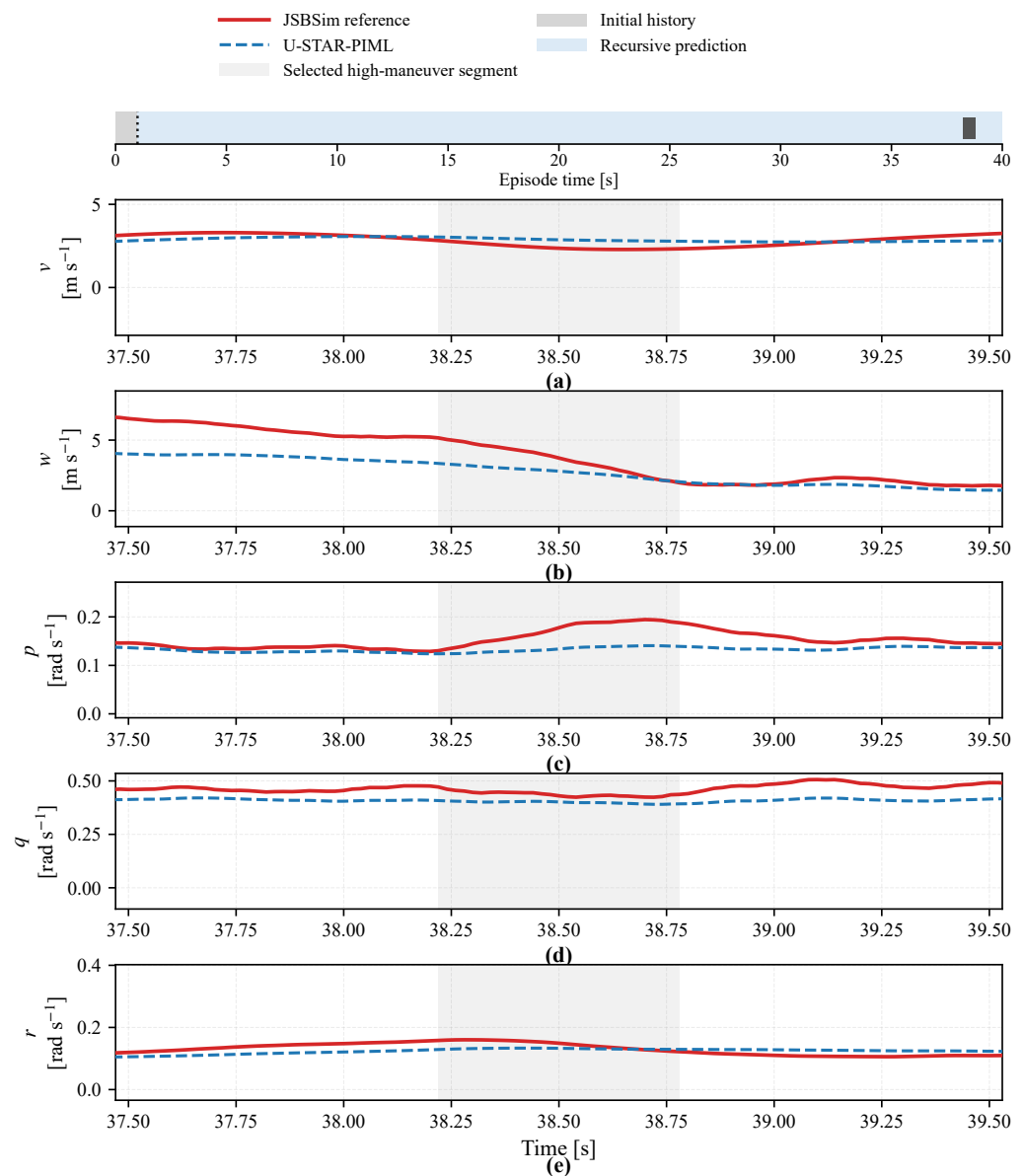


Figure 7. High-maneuver response in wind-OOD episode 63 (38.22–38.78 s), selected by the highest mean maneuver score among 157 retained segments. Panels (a–e) compare v , w , p , q , and r ; shading marks qualifying samples, and the top bar locates the segment within the rollout.

In the displayed segment, U-STAR-PIML captures the overall temporal trends of the five fast states but shows state-dependent discrepancies. The amplitudes of v , p , and q are

visibly underestimated, while the decreasing trend in w is reproduced with a smoother and locally biased response. The prediction of r approaches the reference more closely toward the latter part of the segment. No visible numerical divergence or high-frequency oscillation occurs in this example. Because the segment lies late in a long recursive rollout, the displayed discrepancies include accumulated feedback effects. These local state-wise patterns should not be generalized from the single segment; the aggregate statistics across all detected maneuvers provide the dataset-level assessment.

Across the 157 retained segments from the 10 wind-OOD episodes, the segment-level aggregate fast-state RMSE was 0.774 ± 0.512 (sample SD across segments). State-wise physical-unit errors confirm that the degradation is nonuniform, with the vertical-velocity and pitch-rate channels showing the largest errors within their respective state groups. These fast-state discrepancies can propagate through recursive feedback, but the single displayed segment does not establish a causal mechanism or a common pattern across all maneuvers.

Recursive rollout is a stricter evaluation setting than one-step prediction because predicted states are fed back into the history window and used as inputs for subsequent predictions. Figure 8 visualizes this effect through a three-dimensional trajectory comparison, where velocity and attitude errors can accumulate through the kinematic integration chain and appear as spatial drift.

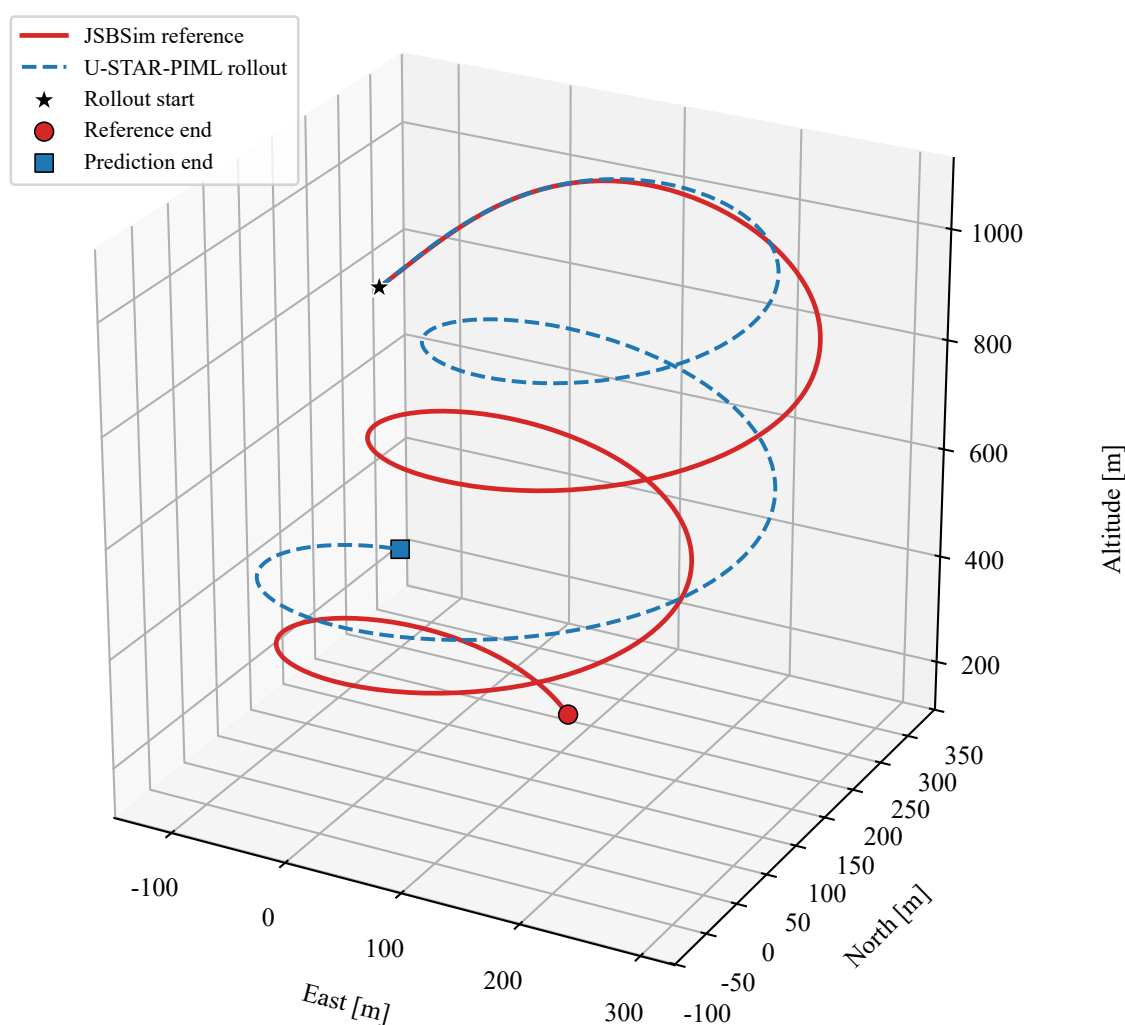


Figure 8. JSBSim reference and recursive U-STAR-PIML trajectory for wind-OOD episode 58 from 1.00 to 40.00 s, following a 50-step warm start. Altitude is $h = -p_d$ in the NED frame.

The three-dimensional trajectory result illustrates accumulated spatial drift rather than isolated one-step discrepancies. Moderate local errors become more visible when each predicted state determines the next history window, so the trajectory view gives a geometric interpretation of horizon-wise rollout error. This distinction is relevant to model-based simulation, trajectory-planning prediction horizons, and prediction-oriented control modules, which repeatedly query a transition model. It does not, however, constitute a closed-loop control or navigation-accuracy validation.

To test whether the position-reference definition affects this conclusion, we repeated the rollout-position evaluation using the direct JSBSim local-position reference, defined as the simulator-provided local position relative to the initial position rather than a quantity reconstructed by velocity integration. Table 6 compares the original kinematically reconstructed reference with this independent reference for E5 and E8. The direct-reference evaluation changes the reported RMSE by at most approximately 0.012%, while the reference disagreement itself is only 0.021209 m for ID and 0.020801 m for wind OOD. The E8 ranking is retained for ID, and the E5 ranking is retained for wind OOD under both reference definitions. This control experiment therefore indicates that the comparative rollout-position conclusion is insensitive to the position-reference definition under the evaluated conditions.

Table 6. Sensitivity of rollout-position RMSE to the position-reference definition.

Model	Condition	Kinematically Reconstructed Reference (m)	Direct JSBSim Reference (m)	Relative Change (%)
E5	ID	18.822916	18.820705	−0.011749
E8	ID	16.924626	16.922836	−0.010579
E5	Wind OOD	72.761558	72.762293	+0.001010
E8	Wind OOD	98.257041	98.256970	−0.000073

Note: The direct reference is the JSBSim local position relative to the initial position, not a ground-truth flight-test position. The reference-disagreement RMSE is 0.021209 m for ID and 0.020801 m for wind OOD.

The horizon-wise rollout error in Figure 9 quantifies how recursive prediction error evolves with prediction horizon. Short horizons mainly reflect local transition accuracy, whereas longer horizons expose error accumulation and free-running stability. This separation is important when a learned model is used for state propagation or a finite prediction window, because one-step accuracy alone cannot guarantee faithful long-horizon trajectories.

In combination with Section 4.2, the horizon-wise trend shows why wind OOD deserves attention in long-horizon prediction. The curve-based comparison separates short-term local behavior from accumulated error and complements the trajectory visualization. The result can inform the choice of prediction horizon in simulation or planning, but it should not be read as evidence that the model improves closed-loop performance.

Table 7 reports the pooled horizon-wise evaluator metric from the five E5 result files. On the ID test set, the aggregate increases monotonically from 0.00427 at 0.2 s to 3.54634 at 20 s, with a marked acceleration beyond 5 s as recursively generated states increasingly determine the history buffer. Wind OOD is separated from the other conditions at every horizon: its error is approximately 15.2 times the ID value at 0.2 s and 5.85 times the ID value at 20 s. The declining ratio does not indicate recovery; the absolute gap expands from approximately 0.061 to 17.21 as both trajectories accumulate error. The parameter, stored-noise diagnostic, and delay conditions remain much closer to ID: their 20 s values differ from the ID aggregate by approximately +6.2%, −6.2%, and −5.3%, respectively, without a consistent degradation ordering across shorter horizons. Because this evaluator metric combines heterogeneous state dimensions, its absolute magnitude

should be interpreted together with the physical-unit grouped position RMSE in Table 5, rather than as a single physically dimensioned error. As in Section 4.2, the stored-noise diagnostic condition remains a control because the stored noisy observations are not consumed by the evaluated model.

Table 7. Pooled recursive rollout RMSE by prediction horizon.

Dataset	0.2 s	1 s	3 s	5 s	10 s	20 s
ID test	0.00427	0.01778	0.06032	0.17680	0.79168	3.54634
OOD wind	0.06502	0.26872	0.85013	2.08054	8.02693	20.75931
OOD parameter	0.00438	0.01623	0.05640	0.15644	0.82262	3.76780
Stored-noise diagnostic	0.00432	0.01609	0.04849	0.14320	0.77089	3.32481
OOD delay	0.00430	0.01789	0.05907	0.14548	0.65620	3.36018

Note: At each horizon, errors are pooled across 10 episodes, valid time samples, and all 12 states. The result is a mixed-unit descriptive metric without a single physical unit. Episode-level values are unavailable, so cross-episode standard deviations cannot be recovered.

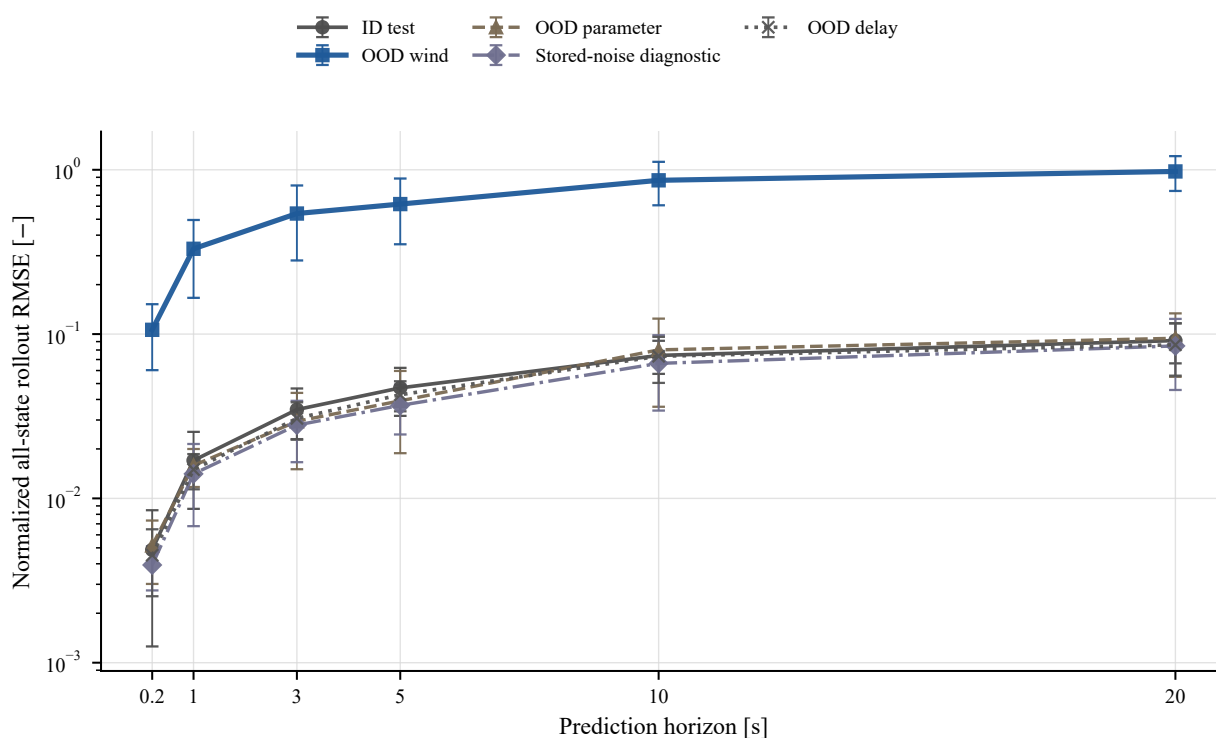


Figure 9. Episode-wise normalized recursive rollout RMSE versus horizon across the ID, prescribed-shift, and diagnostic subsets. Errors are normalized by the training scaler S_x ; curves and error bars show the mean and one sample standard deviation over 10 episodes. The ordinate is logarithmic; Table 7 gives pooled metrics in the original coordinates.

Overall, the dynamic-consistency and rollout analyses show that recursive deployment introduces an error-accumulation problem beyond local one-step prediction. This matters for any simulator or prediction-based decision system that repeatedly feeds model outputs back as inputs. The next subsection turns to one-step predictive uncertainty; uncertainty propagation through recursive rollout is not evaluated and remains an open question.

4.4. One-Step Predictive Uncertainty and Interval Behavior

While Sections 4.2 and 4.3 focus on deterministic prediction accuracy and recursive error accumulation, the proposed model also outputs state-dependent one-step predictive uncertainty. This subsection evaluates the resulting Gaussian intervals across the ID,

effective OOD, and diagnostic conditions. The assessment combines NLL, PICP, MPIW, calibration error, interval visualization, and nominal-versus-observed coverage diagrams.

Figure 10 visualizes the one-step predicted means together with state-dependent Gaussian prediction intervals. It provides a sample-level view of the relation among the target, predictive mean, and interval, complementing the aggregate coverage statistics reported below.

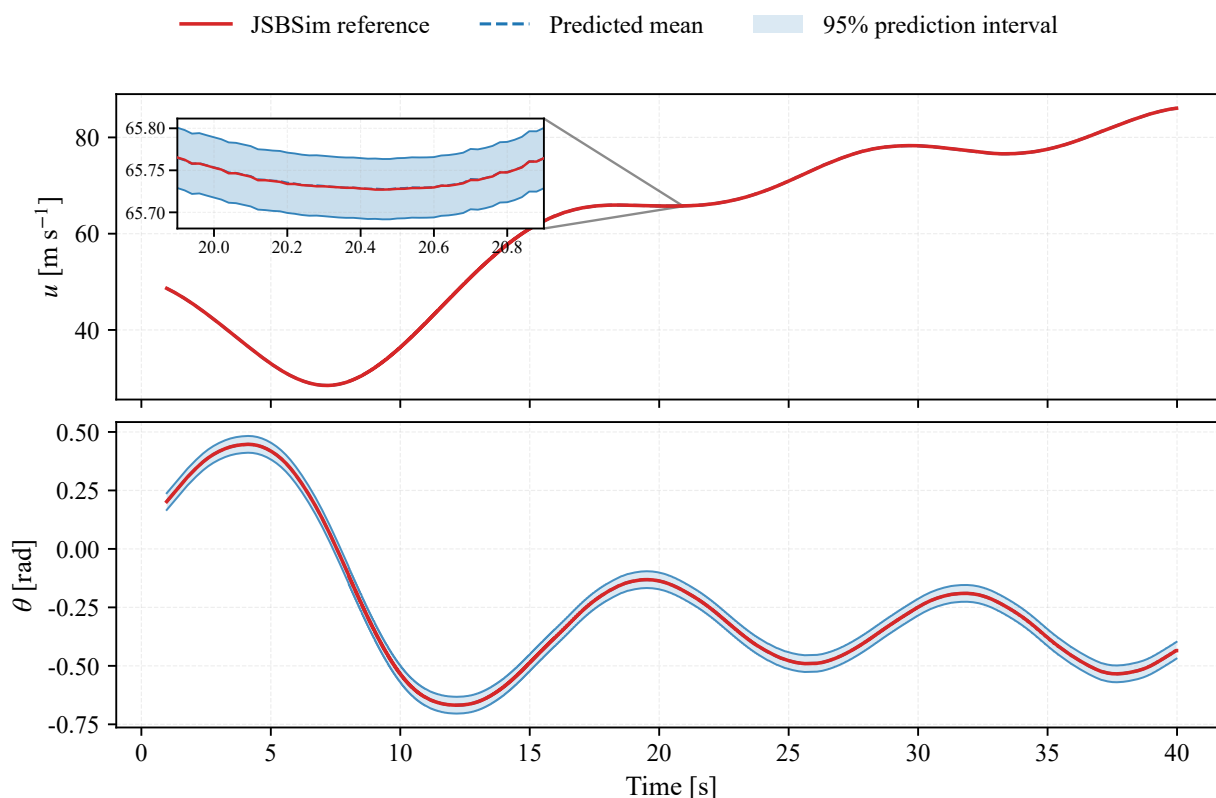


Figure 10. One-step predicted means and state-dependent Gaussian prediction intervals for representative states. The dashed predicted-mean curves may be visually obscured where they closely overlap the JSBSim reference trajectories.

The interval visualization should be interpreted together with PICP and MPIW because coverage alone does not distinguish well-calibrated intervals from conservative ones. This distinction is especially relevant here, as the quantitative results show substantial over-coverage for most test subsets.

Figure 11 further evaluates calibration by comparing nominal and observed coverage for the wind-OOD case. The observed curve remains above the ideal diagonal across the displayed confidence levels, while the gap narrows toward the high-confidence region.

At nominal coverage levels of 90% and 95%, the wind-OOD evaluation achieves observed coverages of 95.40% and 96.68%, respectively. These points lie above the ideal diagonal by 5.40 and 1.68 percentage points. The smaller gap at 95% indicates closer agreement at the upper confidence level, although the intervals remain conservative rather than exactly calibrated.

Table 8 summarizes the one-step uncertainty results. Under ID, the 95% interval coverage is approximately 100%, indicating conservative intervals rather than exact calibration.

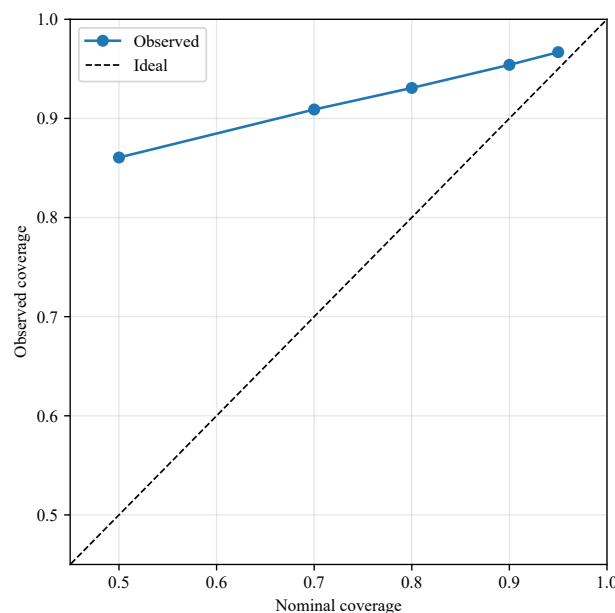


Figure 11. Nominal versus observed coverage of the one-step prediction intervals under the wind-OOD condition.

Table 8. One-step predictive-uncertainty diagnostics.

Dataset	NLL	PICP 90	MPIW 90	PICP 95	MPIW 95	Calib. Error 95
ID test	−3.9786	0.9995	0.06124	1.0000	0.07297	0.04996
OOD wind	−3.6453	0.9540	0.06551	0.9668	0.07806	0.01683
OOD parameter	−3.9790	0.9996	0.06059	0.9999	0.07220	0.04993
Stored-noise diagnostic	−3.9778	0.9996	0.06065	0.9999	0.07227	0.04994
OOD delay	−3.9811	0.9996	0.06060	0.9999	0.07221	0.04994

Note: All metrics use one-step predictions. PICP and calibration error are dimensionless. Aggregate NLL and MPIW combine heterogeneous state coordinates; they are mixed-coordinate evaluator diagnostics rather than a single physically dimensioned uncertainty quantity.

Wind OOD produces the clearest probabilistic response and the largest interval widening. Its 95% empirical coverage moves from approximately 100% under ID to 96.68%, closer to the nominal level, but this change does not indicate more accurate mean prediction; it reflects wider intervals and movement away from coverage saturation under stronger wind disturbance.

The parameter- and delay-OOD cases, together with the stored-noise diagnostic, remain close to the conservative ID behavior. The stored-noise similarity is expected because the modified observation columns are not consumed by the model, so it verifies the data-path boundary rather than uncertainty or sensor-noise robustness.

Overall, the one-step uncertainty output exhibits high empirical coverage and the largest interval widening under wind OOD. Most conditions show conservative over-coverage rather than exact calibration. From an engineering perspective, these state-dependent intervals are best treated as local monitoring cues accompanying the next-state estimate; their conservative behavior may reduce overconfident one-step predictions. The evidence does not establish a trajectory-level risk envelope, certified safety margin, or automatic fallback trigger, and coverage must be interpreted jointly with interval width and deterministic error. The next subsection examines the model components through comparison and ablation.

4.5. Model Comparison and Ablation Study

Sections 4.2–4.4 evaluate E5 in terms of prediction accuracy, recursive stability, and one-step interval behavior. The E1–E8 comparison in this subsection compares outcomes associated with physics guidance, temporal residual learning, gated fusion, staged uncertainty-head training, operator encoding, and rollout fine-tuning under the common ID-test protocol. Because the configurations are not matched in every architectural and training-budget dimension, the comparison supports association and trade-off analysis rather than causal attribution to a single component. The comparison also distinguishes local prediction accuracy from recursive stability, probabilistic output, and computational cost.

Figure 12a,b show the representative seed-42 comparison. E1 has a one-step RMSE of 0.111880, whereas the learned variants remain near the 0.0055 range; its representative rollout-position error is also far larger than that of the learned models (for example, 431.79 m for E1 versus 20.85 m for E3). Across the one-step, rollout-position, and 20 s metrics, the learned variants therefore show approximately 94–98% lower errors than E1 in this representative comparison. E1 was not included in the five-seed analysis, so this contrast is a descriptive single-run baseline comparison rather than evidence of superiority over a fully identified high-fidelity physics model. The compact prior preserves physical structure but lacks sufficient fidelity for the unresolved aerodynamic, disturbance, and simulator-specific dynamics in this benchmark, supporting learned residual correction without implying conclusions about certified flight modeling.

In the representative seed-42 run, E3 retains similar local accuracy to E2 while reducing rollout-position RMSE from 25.71 to 20.85 m. Its 20 s RMSE is also lower (2.640 versus 3.553), although this seed-42 minimum is not a repeated-run ranking. These observations motivate the residual branch for recursive consistency, while the five-seed comparison below provides the primary evidence for overall ranking.

E4 preserves the local accuracy of E3 but does not improve either recursive metric in the ID-only ablation. The comparison therefore provides no direct evidence of a deterministic rollout gain from gating, although the gate may serve a different role under varying or OOD conditions. Table 9 does not isolate that contribution.

The learned-gate characterization provides an empirical description of this role. The final global physics coefficient was $\tau = 0.156862$, which did not collapse to zero but was substantially lower than its initialization of 0.35. The physics branch therefore remained active, although its scalar contribution was down-weighted. The six-state mean residual-contribution fractions, computed from the mean absolute branch contributions rather than from τ alone, were 57.6% for ID, 57.3% for wind OOD, and 57.5% for parameter OOD. Except for u , the effective physics and residual contributions were broadly comparable in most dynamic channels. Thus, τ should not be interpreted as the percentage contribution of the physics branch. Under ID, the mean α_t values for (u, v, w, p, q, r) were (0.999943, 0.999551, 0.998899, 0.999983, 0.999875, 0.999879), and parameter OOD showed the same near-fully open pattern, (0.999876, 0.999605, 0.998953, 0.999980, 0.999891, 0.999865).

Wind OOD activated stronger and uneven state-dependent residual modulation across the ten evaluated episodes. The mean gate values were $u = 0.988317$, $v = 0.971110$, $w = 0.909010$, $p = 0.994318$, $q = 0.974340$, and $r = 0.977636$. The corresponding standard deviations were 0.044000, 0.089715, 0.175155, 0.027090, 0.099239, and 0.069551, and the minima were 0.321988, 0.087593, 0.083242, 0.408961, 0.071518, and 0.363204, respectively. The most pronounced variation occurred in w , followed by q , v , and r . The representative heatmap in Figure 13 illustrates pronounced w and q modulation around 9–15 s, without being used to generalize beyond that episode. Relative to the low-wind interval, the high-wind mean gate changes were -0.017067 (u), -0.044974 (v), -0.125510 (w), -0.010410 (p), -0.046235 (q), and -0.039794 (r). A lower α_t indicates suppression of the residual

correction rather than a direct decrease in physics trust. Spearman correlations between α_t and wind magnitude were -0.355246 (u), -0.141311 (v), -0.375672 (w), -0.345311 (p), -0.399591 (q), and -0.293004 (r). The corresponding correlations with raw physics mismatch were 0.144642 , 0.262004 , 0.430499 , 0.073019 , 0.382312 , and -0.079727 . The stronger positive associations for w and q are consistent with state-specific sensitivity to local discrepancy, but they do not establish causality or a simple monotonic transfer of trust away from physics.

Table 9. ID-test model comparison and ablation results.

Experiment	Model Variant	One-Step RMSE	Rollout Pos. RMSE	20 s RMSE	PICP 95	Runtime
E1	Physics-only baseline	0.111880	431.79	113.291	–	0.0331
E2	Data-driven baseline	0.005538	25.71	3.553	–	0.0403
E3	Physics-residual model	0.005566	20.85	2.640	–	0.0301
E4	State-dependent gated fusion model	0.005557	21.00	2.852	–	0.0335
E5	Full U-STAR-PIML	0.005513	18.82	3.546	1.0000	0.0266
E6	Operator data model	0.005591	19.42	2.723	–	0.0434
E7	Operator residual model	0.005651	22.25	5.633	–	0.0315
E8	Operator rollout model	0.005659	16.92	2.726	–	0.0513

Note: Metrics share the ID split and protocol; PICP 95 applies only to variants with the one-step uncertainty head, and runtime is mean one-step time in milliseconds. Table 9 reports the original representative seed-42 comparison. Five-seed reproducibility of E2–E8 is evaluated separately in Table 10 and Figure 14. Unequal method-specific training budgets remain and prevent architecture-only causal attribution. E2 (GRU) and E6 (FNO) are data-driven references; TCN and transformers are not evaluated.

In the representative seed-42 run, E5 combines a low one-step RMSE (0.005513), low rollout-position RMSE (18.82 m), and a conservative 95% interval coverage of 1.0000. Its position score is second to E8 in that run, while its broader 20 s all-state score reflects a different objective. Because no matched E5-without-uncertainty-head or end-to-end E5 variant is available, these differences cannot be attributed to a single training component.

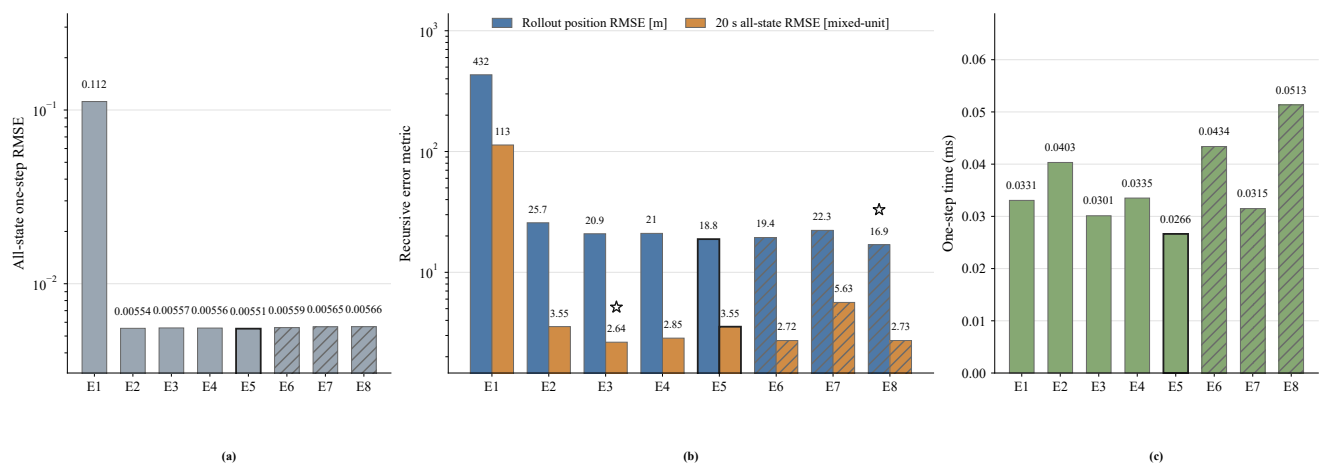


Figure 12. Representative seed-42 ID-test comparison of sequential variants: (a) one-step RMSE; (b) rollout-position and 20 s all-state RMSE; and (c) runtime on an NVIDIA RTX 4070 graphics processing unit (GPU) with batch size 256. Panels (a,b) use logarithmic scales. Stars mark seed-42 minima at least 3% below the second-lowest value; black outlines mark E5, and hatching marks E6–E8. Five-seed variability of the learned variants is reported separately in Table 10 and Figure 14.

In the representative seed-42 run, the operator variants show metric-dependent effects: E6 improves both recursive metrics relative to E2, whereas adding the operator residual

formulation in E7 does not preserve that behavior. These observations are limited to the current architectures, optimization schedules, and representative run and are not treated as overall repeated-run rankings.

In the representative seed-42 run, E8 attains a rollout-position RMSE of 16.92 m, illustrating that suppressing position drift and reducing all-state long-horizon error are related but nonequivalent objectives. The five-seed analysis does not support a universal operator advantage, and the representative timing profile cannot attribute any difference causally to rollout fine-tuning.

Table 10. Five-seed reproducibility of the learned variants on the ID-test set.

Experiment	One-Step RMSE	Rollout-Position RMSE (m)	20 s All-State RMSE
E2	0.005537 ± 0.000005	21.51 ± 3.88	2.923 ± 0.623
E3	0.005558 ± 0.000012	20.54 ± 1.85	2.996 ± 0.264
E4	0.005549 ± 0.000008	20.43 ± 3.81	3.175 ± 0.392
E5	0.005515 ± 0.000009	18.61 ± 2.15	3.067 ± 0.387
E6	0.005601 ± 0.000007	21.08 ± 3.29	3.304 ± 1.100
E7	0.005652 ± 0.000020	23.50 ± 1.94	3.953 ± 1.508
E8	0.005650 ± 0.000010	21.92 ± 4.39	3.946 ± 1.095

Note: Values are the sample mean $\pm$ sample SD over five independent training seeds (42, 142, 242, 342, and 442). The episode split is fixed across runs. Models retain their original method-specific optimization schedules; therefore, this table characterizes run-to-run reproducibility rather than matched-budget architecture effects.

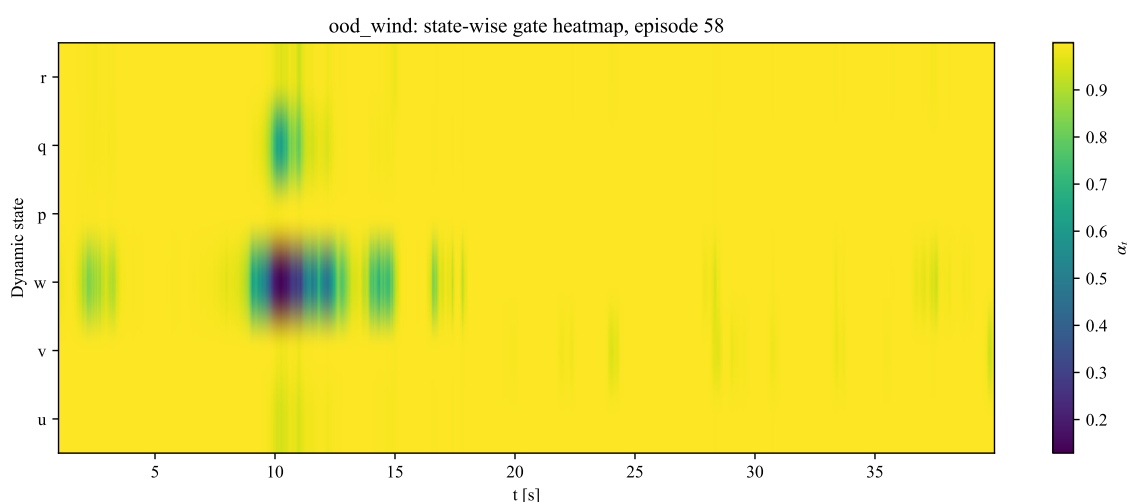


Figure 13. State-wise residual-gate response of E5 under the representative wind-OD episode 58. The six rows correspond to the dynamic states u , v , w , p , q , and r . Values near one indicate that the learned residual correction is passed with little attenuation, whereas lower values indicate stronger suppression of the residual contribution. The most pronounced modulation occurs in the vertical-velocity and pitch-rate channels during the strong transient region. The gate should not be interpreted as a normalized physics-versus-data mixing coefficient.

Figure 12c shows the representative seed-42 runtime profile. E5 requires 0.0266 ms per one-step forward pass on an NVIDIA RTX 4070 with batch size 256. This desktop-GPU measurement suggests that neural inference is unlikely to dominate computation in the reported setting, but it is not evidence of real-time onboard capability. Embedded-hardware tests, worst-case latency, memory, energy, and closed-loop scheduling measurements would be required for that claim; the profile is therefore retained as a representative profiling result rather than a stable runtime ranking.

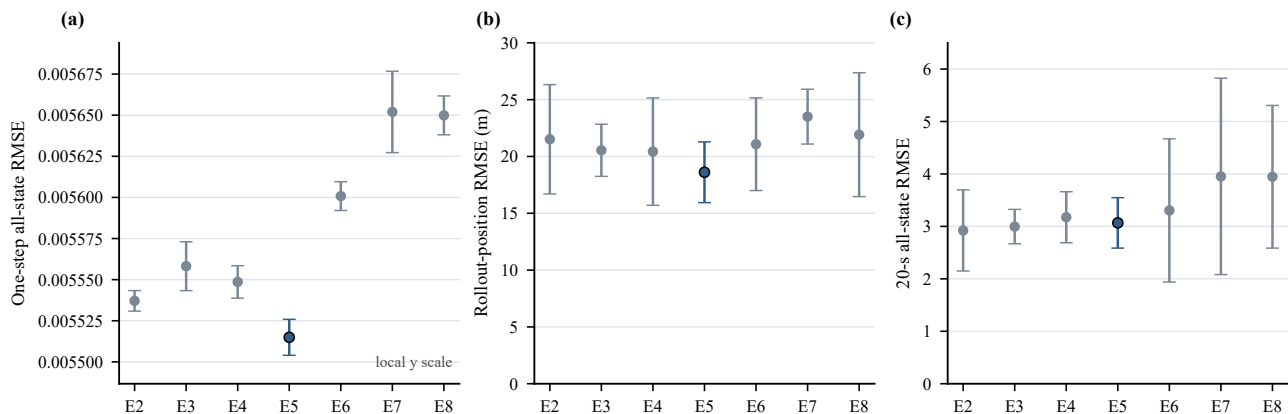


Figure 14. Five-seed reproducibility of the learned variants under their original method-specific training schedules. Markers denote sample means, and error bars denote 95% Student's *t* confidence intervals over seeds 42, 142, 242, 342, and 442. The episode split is fixed across runs. Panel (a) uses a labeled local vertical scale to resolve the small one-step differences; panels (b,c) use linear scales beginning at zero. The analysis quantifies run-to-run variability and does not constitute a matched-budget architecture-only comparison.

Five-Seed Reproducibility Analysis

Table 10 and Figure 14 summarize the repeated-run results computed from the 35 completed runs. E1 is excluded because it was retained only as the original single-run compact-physics reference and was not retrained across the five seeds.

Across the five runs, E5 achieves the lowest mean one-step RMSE, 0.005515 ± 0.000009 , compared with 0.005537 ± 0.000005 for E2. E5 produces the lower one-step RMSE in all five paired seeds. The mean paired difference, E2 minus E5, is 2.22×10^{-5} (95% paired Student's *t* confidence interval: 1.65×10^{-5} – 2.79×10^{-5}), corresponding to approximately a 0.40% relative reduction. The repeated-run analysis therefore supports a consistent but quantitatively small local-accuracy advantage rather than a large practical separation. For model selection, this reproducible local difference should be weighed against the intended prediction horizon: short-horizon applications may value stable one-step transitions, whereas longer-horizon prediction must also account for position drift and run-to-run variability.

E5 also has the lowest five-seed mean rollout-position RMSE, 18.61 ± 2.15 m. However, the lowest-error variant changes from E8 to E4, E5, E5, and E2 across seeds 42, 142, 242, 342, and 442, respectively, and the confidence intervals are broad. The repeated-run results therefore do not support a universal winner for recursive position accuracy. For the 20 s all-state metric, E2 has the lowest mean, 2.923 ± 0.623 , while E3 and E5 remain close at 2.996 ± 0.264 and 3.067 ± 0.387 , respectively. The lowest-error variant is E3, E7, E2, E2, and E2 across the same five seeds, so the analysis does not establish a stable long-horizon winner among the leading variants. The operator-based variants E6–E8 also exhibit larger run-to-run dispersion in the 20 s metric than E3 and E5 under the current training protocols.

Overall, E1 remains a single-run reference showing that the isolated compact physics prior is insufficient for accurate simulator-dynamics prediction. Among E2–E8, the five-seed analysis confirms a reproducible but small one-step advantage for E5. E5 has the lowest mean rollout-position error, and E2 has the lowest mean 20 s all-state error, but both recursive rankings vary across seeds. The repeated-run evidence therefore supports objective-dependent and seed-sensitive trade-offs rather than a universal model winner.

4.6. External Real-Flight Zero-Shot Validation

The frozen E5 checkpoint was evaluated on the external IDF-DS telemetry [39] using the protocol described in Section 4.1. The comparison used the current state as the next-

state prediction for the Persistence baseline and did not include fine-tuning, test-time adaptation, or recursive rollout. Table 11 reports pooled grouped RMSE over 29,500 valid one-step samples from 10 independent logs. The grouped values combine states with different physical units and are therefore descriptive, consistent with the existing grouped-RMSE convention.

Table 11. Zero-shot one-step performance on external real-flight data. RMSE values combine states with different physical units and therefore serve only as descriptive grouped metrics, consistent with this manuscript’s existing grouped-RMSE convention.

State Group	E5 Zero-Shot RMSE	Persistence RMSE
Body velocity	0.05033	0.04831
Angular rate	0.04500	0.04546
Attitude	0.000904	0.005867
Position	0.002573	0.238353
All-state	0.03379	0.12374

The pooled all-state RMSE is lower for E5 than for Persistence (0.03379 versus 0.12374), corresponding to an approximately 72.7% reduction; the attitude and position reductions are approximately 84.6% and 98.9%, respectively. The body-velocity group does not improve (0.05033 versus 0.04831), so the state-group ranking is not uniformly favorable and body-velocity prediction shows no consistent flight-level advantage. The transfer benefit is concentrated in attitude and position propagation, whereas several fast dynamic channels remain comparable to, or slightly worse than, Persistence. At the flight level, E5 yields a lower pooled all-state one-step RMSE than Persistence on all 10 independent flights. The paired mean difference, Persistence minus E5, is 0.0895 with a 95% confidence interval of 0.0808–0.0983, and all 10 paired differences are positive.

Figure 15 visualizes this flight-wise consistency. The external trajectories also exhibit substantial distribution mismatch relative to the C172x training data, particularly in lateral/vertical velocity and angular-motion channels, reinforcing that the experiment represents a nontrivial cross-airframe transfer setting. These results provide an external transfer diagnostic for one-step prediction rather than evidence of successful general sim-to-real transfer: the real airframe, telemetry-derived state estimates, actuator mapping, and onboard wind estimate differ from the training environment, and no recursive real-flight stability, same-airframe validation, or closed-loop operational performance was assessed.

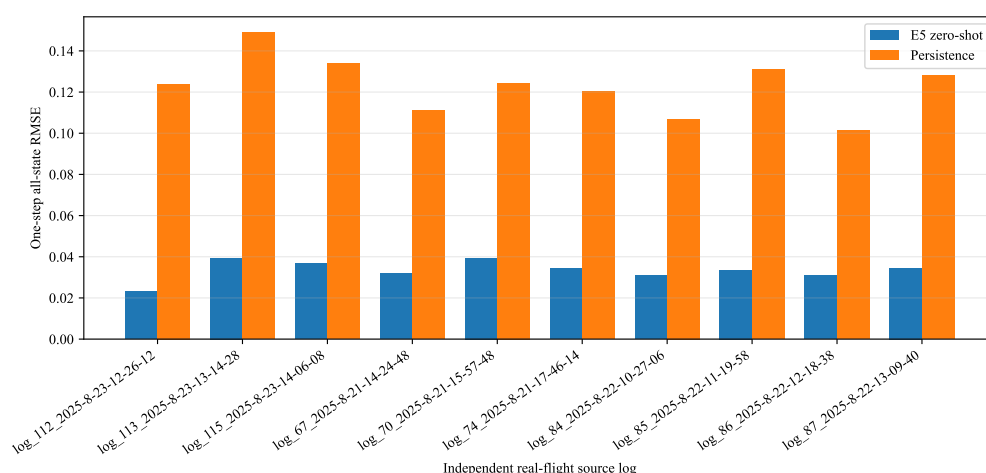


Figure 15. Flight-wise zero-shot one-step all-state RMSE on 10 independent IDF-DS Ranger 2400 real-flight logs. E5 was trained only on JSBSim C172x data and was evaluated without fine-tuning or test-time adaptation. Persistence uses the current state as the next-state prediction.

5. Discussion

The case study in Section 4 reveals three findings that shape the interpretation of U-STAR-PIML. Learned temporal models produce lower prediction errors than the isolated compact 6DoF baseline, wind OOD produces the strongest degradation among the prescribed shifts under the simulator-provided known-wind protocol, and different evaluation objectives favor different model variants. In particular, local transition accuracy, all-state long-horizon accuracy, position-drift suppression, probabilistic output, and computational cost do not induce a single model ranking. The following discussion therefore examines the mechanisms underlying physics–residual learning, wind-induced error propagation, the scope of the one-step uncertainty output, and long-horizon model selection trade-offs rather than restating the case study results.

5.1. Interpretation of Physics–Residual Learning and Gated Fusion

The separation between E1 and the learned variants in Table 9 and Figure 12 shows that the compact physics branch is insufficient as an isolated predictor. E1 has a one-step RMSE of 0.111880, whereas E2–E8 remain near 0.00551–0.00566, and its recursive errors are also substantially larger. The branch preserves rigid-body organization but omits aerodynamic, propulsion, disturbance, actuator, and simulator-specific effects. This result supports treating compact physics as a structural prior complemented by learned correction, consistent with physics-based and residual model-error formulations [2,8,23], rather than as evidence that physical guidance is unhelpful.

The representative seed-42 run suggests improved recursive consistency for E3 relative to E2. Across five seeds, however, E3 retains a modestly lower mean rollout-position error (20.54 versus 21.51 m) but does not reduce the mean 20 s all-state RMSE (2.996 versus 2.923). Its mean one-step RMSE is also slightly higher (0.005558 versus 0.005537). The effect of residual physics guidance is therefore metric-dependent and should not be interpreted as a uniform improvement in recursive stability.

The E3–E4 comparison provides a narrower interpretation of gated fusion. Across five seeds, E4 preserves similar local accuracy to E3 (0.005549 versus 0.005558) and yields a slightly lower mean rollout-position RMSE (20.43 versus 20.54 m), while its mean 20 s all-state RMSE increases from 2.996 to 3.175. Accordingly, the current comparison does not establish a uniform recursive advantage from gated fusion. The learned-gate analysis instead provides evidence that the gate performs state- and condition-dependent residual modulation, particularly under wind OOD. The final global physics coefficient was $\tau = 0.156862$, which remained nonzero, while the mean residual-contribution fractions were approximately 57.6%, 57.3%, and 57.5% for ID, wind OOD, and parameter OOD, respectively. These effective contributions are computed from $|\tau \dot{\mathbf{x}}_{\text{phys}}|$ and $|\alpha_t \odot \dot{\mathbf{x}}_{\text{res}}|$; they cannot be inferred from τ alone.

The gate is therefore best interpreted as condition-dependent residual regulation rather than normalized trust transfer. It remains close to fully open under ID and parameter OOD, whereas wind OOD produces stronger and uneven state-specific modulation, especially in the w and q channels. The negative association with wind magnitude and the positive association of several channels with local physics mismatch are consistent with state-dependent modulation, but they do not establish causality or imply that increasing wind automatically reduces physics trust. The operational description “trust-aware gated fusion” therefore refers to this condition-dependent capacity, not to sustained redistribution under every evaluated condition.

The deterministic differences between E4 and E5 cannot be assigned to staged training or uncertainty-head learning individually. E5 combines staged optimization, deterministic joint fine-tuning, recursive supervision, continued uncertainty-only training, and associated

loss scheduling, while no matched end-to-end E5 or E5-without-uncertainty-head variant is reported. Separating these tasks is methodologically consistent with known gradient imbalance, unequal convergence, and ill-conditioned optimization in physics-informed learning [9,10,22]. Nevertheless, the present ablation does not independently verify the benefit of any single E5-specific training phase.

5.2. Wind-Induced Distribution Shift and Error Propagation

Among the prescribed shifts, wind OOD produces the largest degradation in Table 5 and Figure 5 under the simulator-provided known-wind protocol. Relative to ID, the body-velocity and angular-rate RMSE increase by factors of approximately 8.90 and 6.50, while rollout-position RMSE increases from 18.82 to 72.76, a factor of approximately 3.87. Stronger wind directly changes relative airflow and therefore the aerodynamic forces and moments driving fast states, which provides a physically plausible mechanism for the observed sensitivity. The ranking nevertheless applies only to the selected parameter ranges: the shift severities are expressed in incommensurable units and are not calibrated to a common departure from nominal. Parameter and delay remain closer to ID under this protocol. The stored-noise diagnostic is excluded from this OOD ranking because it does not alter the clean-state network input.

The derivative, maneuver, trajectory, and horizon results indicate a physically coherent error-propagation path. Wind-driven airflow variation perturbs velocity and angular-rate derivatives, as examined in Figures 6 and 7; the resulting attitude deviations then enter kinematic integration and appear as spatial drift in Figure 8. In the pooled evaluator metric of Table 7, wind-OOD error increases from 0.0650 at 0.2 s to 20.759 at 20 s, compared with an ID 20 s value of 3.546. The physical-unit rollout-position RMSE separately increases from 18.82 to 72.76. Position error is therefore a downstream accumulation of coupled dynamic and attitude errors rather than the consequence of one state variable alone.

This mechanism motivates broader wind-focused evaluation rather than an immediate architectural conclusion. Future studies could randomize wind magnitude, direction, gust frequency, and temporal spectrum more extensively; represent relative airspeed or aerodynamic disturbance more directly, as demonstrated by residual aerodynamic learning under strong winds [24]; and adaptively sample wind segments with high residuals. Combined wind-parameter-noise-delay shifts would test interactions hidden by single-factor subsets. Residual- or uncertainty-informed gating could also examine whether physics trust should decrease when local physical inconsistency rises, but the current experiments do not establish that such changes would improve robustness.

5.3. Interpretation and Scope of the One-Step Predictive Uncertainty Output

The one-step uncertainty head defines a conditional heteroscedastic Gaussian variance for each state, based on the available history and current inputs. It supplies state-wise intervals around the next-state mean, as illustrated in Figure 10. This definition describes local predictive dispersion for a single transition rather than a complete representation of all uncertainty sources.

In engineering terms, the output is an auxiliary indication of local next-state dispersion under the current model and input conditions. It can support prediction monitoring or conservative downstream decision logic, but these uses remain prospective. The present study does not validate an automatic fallback trigger, controller, risk threshold, certification criterion, or safety bound.

Table 8 and Figure 11 show that interval coverage must be interpreted together with likelihood and width. ID, parameter OOD, the stored-noise diagnostic condition, and delay OOD have 95% coverage near 0.9999–1.0000, indicating systematic over-coverage and

conservative intervals. Wind OOD moves coverage to 0.9668 and calibration error to 0.01683, while NLL changes to -3.6453 and the 95% interval widens to 0.07806. This movement toward nominal coverage does not imply a more accurate deterministic prediction or generally superior uncertainty estimation; it reflects wider intervals and reduced coverage saturation under the wind-shift condition.

The output does not represent epistemic or model-parameter uncertainty because the model uses no Bayesian neural network, deep ensemble, or Monte Carlo dropout. It is also not a complete aleatoric/epistemic decomposition. Because the network reads clean JSBSim states rather than the stored noisy observations, Σ_t is not sensor, IMU, GNSS, or measurement-covariance uncertainty [11,27].

The assessment remains restricted to one-step metrics. The recursive rollout feeds back only deterministic means and does not propagate samples or covariance. The intervals therefore do not establish stochastic recursive propagation, trajectory-level coverage, horizon-dependent calibration, or a probability that an aircraft trajectory remains inside an envelope. No conformal guarantee, formal confidence bound, flight-safety certification, or other safety guarantee is claimed.

Table 9 also lacks a matched E5-without-uncertainty-head model, so the head's independent influence on deterministic RMSE cannot be isolated. Future work could examine ensembles, Bayesian approximations, conformal recalibration, low-rank covariance, and Monte Carlo rollout with horizon-wise and trajectory-level evaluation. These capabilities are not part of the present method and should be evaluated under matched computational budgets.

5.4. Long-Horizon Trade-Offs and Model Selection Implications

Table 10 and Figure 14 strengthen the distinction among the evaluation objectives. E5 has the lowest mean one-step RMSE and is lower than E2 in all five paired seeds, but the relative mean reduction is only approximately 0.40%. E5 also has the lowest mean rollout-position RMSE, although the lowest-error variant changes across seeds, and the confidence intervals are broad. E2 has the lowest mean 20 s all-state RMSE, while E3 and E5 remain close in mean performance. Run-to-run variation is sufficiently large that the repeated-run analysis does not establish a stable recursive-metric winner among these leading variants.

The operator variants further separate position and all-state objectives. In the representative seed-42 run, E6 improves both recursive metrics relative to E2, and E8 attains the lowest rollout-position error. Across five seeds, E6 retains only a slightly lower mean position error than E2 (21.08 versus 21.51 m) and has a higher mean 20 s error (3.304 versus 2.923). E8 does not retain the lowest mean rollout-position error. Moreover, the 20 s SDs of E6–E8 range from 1.095 to 1.508, compared with 0.264 for E3 and 0.387 for E5. This dispersion is specific to the present architectures and optimization protocols and does not establish inherent instability of operator learning [25,26].

Model selection should consequently follow the intended deployment objective and account for optimization stochasticity. Local one-step ranking is highly reproducible, whereas recursive position and long-horizon all-state rankings are considerably more seed-sensitive. E5 provides the lowest one-step and rollout-position sample means together with probabilistic intervals, while E2 provides the lowest 20 s sample mean. None of these sample-mean rankings alone establishes architecture-level superiority because the method-specific training budgets remain unequal. The seed-42 runtime profile is retained as a representative measurement and does not establish a stable runtime ranking or real-time capability.

5.5. Limitations and Future Work

An external zero-shot evaluation on 10 independent IDF-DS Volantex Ranger 2400 flight logs provides an initial check beyond the JSBSim environment. The consistent reduction in pooled all-state one-step error indicates that part of the learned transition structure transfers to real fixed-wing telemetry despite the simultaneous simulator-to-real and cross-airframe shift. However, the experiment remains limited to one-step prediction on a public external dataset, and several fast dynamic channels do not improve over Persistence. It therefore does not establish same-airframe sim-to-real transfer, onboard closed-loop performance, or operational robustness. Simulator-specific dynamics may still differ from the sensor, actuator, structural, and atmospheric effects encountered on a physical unmanned fixed-wing aircraft, and cross-airframe transfer remains incompletely characterized. Future work should therefore evaluate same-airframe real-flight datasets, recursive real-flight rollouts, hardware-in-the-loop experiments, additional unmanned fixed-wing configurations, cross-airframe adaptation, and simulator-to-real transfer before drawing conclusions about operational UAV performance.

Accordingly, the core model-development evidence establishes simulation-domain methodological validity, while the external evaluation provides a bounded transfer diagnostic rather than operational validity on a physical UAV. Within the simulated setting, the physics-residual formulation is designed to accommodate selected model-form discrepancies and operating-condition variations represented in the training data, while recursive-rollout supervision targets accumulated prediction error and state-dependent residual gating provides a mechanism for history- and state-varying correction. The external result suggests that part of this transition structure transfers, but the present evidence does not establish compensation for noisy IMU, GNSS, or air-data-derived state estimates, sensor bias, dropout or latency, unknown or misestimated wind, realistic actuator nonlinearities, saturation or dead zones, structural flexibility or vibration, faults, icing, damage, unseen airframe dynamics, cross-airframe mismatch, real embedded-computing behavior, or closed-loop flight-control performance. Domain randomization is therefore used here to reduce sensitivity to selected simulated variations rather than as evidence of general simulator-to-real transfer.

The OOD design uses predefined single-factor subsets. Wind, parameter, and delay alter the model inputs or underlying dynamics, whereas the stored-noise multiplier changes only unused observation columns. The stored-noise diagnostic therefore does not constitute a sensor-noise robustness experiment because the evaluated model consumes clean simulator states. In an operational UAV, a dynamics predictor would normally receive states estimated from IMU, GNSS, air-data, or other onboard measurements. Errors, bias, dropout, and latency in those state estimates constitute a distinct robustness problem that is not evaluated here. Robustness to noisy or intermittent onboard state estimates therefore remains an explicit limitation of the present simulation protocol. Future work should evaluate physically grounded measurement errors or an explicit state-estimation pipeline without treating the current diagnostic as evidence for those conditions.

The configured severities are expressed in incommensurable units—wind statistics, dimensionless scale factors, a stored-noise multiplier, and a delay in simulation steps—and are not calibrated to a common relative departure from nominal. The observed ranking therefore reflects the particular effective shifts and diagnostic setting used here rather than an intrinsic ordering of the underlying mechanisms. Combined effective shifts, continuous severity changes, structural faults, icing, sensor failures, and unseen maneuvers are not represented, so robustness outside the reported subsets remains unknown. Future evaluation should include graded-severity sweeps, compositional OOD tests, fault and icing scenarios,

and progressively increasing shift severity. Unknown-shift detection could be coupled with one-step interval monitoring in later work, but it is not part of the present contribution.

The probabilistic model uses diagonal one-step transition variance and does not recursively propagate samples or covariance. Cross-state covariance, horizon-wise interval behavior, and trajectory-level calibration therefore remain unresolved, and the output should not be interpreted as capturing every source of process and model uncertainty. Low-rank or full covariance, stochastic rollout, ensembles, Bayesian approximations, and conformal calibration provide possible extensions. Their value should be judged through both calibration and deterministic accuracy rather than coverage alone.

Component attribution remains limited because no E5-without-uncertainty-head model, matched staged-versus-end-to-end E5 comparison, or complete gate ablation across all OOD splits is reported. The five-seed analysis quantifies run-to-run variability under the original method-specific schedules, but it does not remove unequal optimization budgets or isolate architecture from training protocol. A fully matched-budget multi-seed comparison was not completed; consequently, the repeated-run evidence establishes reproducibility under the reported schedules rather than architecture-only effects. The learned gate is now directly characterized, but its values remain close to saturation under ID and parameter OOD, while stronger modulation is observed mainly under wind OOD and remains uneven across channels. Future work should use fully matched budgets, gate-intervention tests, stage and loss sensitivity analyses, and controlled parameter-count comparisons. Deployment studies should additionally quantify memory, energy, worst-case latency, and closed-loop behavior before considering quantization, pruning, or onboard integration.

Overall, the results indicate that learned temporal correction improves prediction relative to the incomplete compact physics prior under the present simulation protocol. The repeated-run analysis supports a consistent but small E5 one-step advantage, while the recursive effects of residual guidance and operator extensions remain metric and seed-dependent. Different variants offer distinct local, all-state long-horizon, position-drift, and probabilistic capabilities. Among the prescribed shifts, wind OOD produces the largest degradation under the simulator-provided known-wind protocol. The uncertainty conclusions apply only to one-step prediction, while the external real-flight result provides an initial transfer diagnostic rather than evidence of same-airframe sim-to-real validity.

6. Conclusions

This study developed U-STAR-PIML, a recursive one-step dynamics learning framework for 12-state fixed-wing UAV flight dynamics prediction. The framework integrates a learnable compact 6DoF physics prior, a history-dependent temporal residual branch, trust-aware fusion with state-dependent residual gating, hard attitude and position kinematics, and a state-dependent one-step uncertainty head. A six-stage procedure separates deterministic drift learning from uncertainty-head training, with interval calibration assessed post hoc rather than optimized through a separate calibration loss. The framework and its intended applications are restricted to unmanned fixed-wing aircraft, while the C172x model serves only as a surrogate six-degree-of-freedom simulation benchmark and is not considered the target operational platform. For this unmanned-aircraft prediction task, a reproducible ID and prescribed-shift protocol, together with a stored-noise diagnostic control, provided a common basis for assessing one-step accuracy, recursive rollout stability, and one-step interval behavior.

Four main conclusions follow from the case study. First, the representative seed-42 comparison shows lower one-step, rollout-position, and 20 s errors for the learned temporal variants than for the isolated compact physics prior. Because E1 was not included in the repeated-run analysis, this comparison is treated as a descriptive single-run baseline result

rather than a five-seed estimate. Across five seeds, E5 yields the lowest mean one-step RMSE and is lower than E2 in every paired run, although the mean relative reduction is only approximately 0.40%. Second, among the prescribed shifts, wind produces the largest degradation under the known-wind protocol, in which the exact wind vector used in the JSBSim simulation is supplied to the model at every prediction step. The stored-noise diagnostic does not alter the clean-state network input and is not interpreted as evidence of sensor-noise robustness. Third, the one-step uncertainty head produces conservative interval behavior under most evaluated conditions, and this evidence does not establish trajectory-level calibration. Fourth, E5 has the lowest five-seed mean rollout-position RMSE and E2 has the lowest five-seed mean 20 s RMSE, but both recursive rankings vary across seeds. Model selection should therefore follow the deployment objective and account for training stochasticity. The learned gate shows its strongest state-dependent modulation under wind OOD, while the direct-position-reference control confirms that the representative seed-42 E5/E8 rollout-position ordering is insensitive to the reconstruction procedure under the evaluated conditions.

The principal contribution of this work is methodological: hybrid physics–residual dynamics learning, trust-aware gated fusion with state-dependent residual modulation, recursive rollout supervision, and state-dependent one-step predictive variance are organized within one staged formulation. The reproducible ID and prescribed-shift protocol, together with the stored-noise diagnostic control, treats local accuracy, recursive stability, and one-step interval behavior as distinct evidence categories. The resulting framework and evaluation design are applicable to model-based simulation, state prediction, and prediction-oriented analysis of unmanned fixed-wing platforms in which the available physical model is informative but incomplete.

External zero-shot evaluation on 10 independent Ranger 2400 real-flight logs provides initial evidence that the learned one-step transition structure can transfer beyond the JSBSim C172x environment, with E5 reducing the pooled all-state RMSE from 0.12374 for Persistence to 0.03379. This result is limited to one-step external telemetry evaluation and does not establish same-airframe sim-to-real transfer, recursive real-flight stability, or closed-loop operational validity. The present conclusions establish simulation-domain methodological validity and an external transfer diagnostic but remain restricted to the evaluated JSBSim and IDF-DS conditions, a diagonal one-step transition variance, and partially coupled comparisons, as detailed in Section 5.5. Future work will accordingly prioritize same-airframe real-flight and hardware-in-the-loop validation, recursive real-flight evaluation, trajectory-level uncertainty propagation and calibration, compositional and progressively severe distribution shifts, and fully matched-budget multi-seed ablations that isolate the individual contributions of gating, staged training, and the uncertainty head.

Author Contributions: Conceptualization, Z.G. and Z.Z.; methodology, Z.G. and Z.Z.; software, Z.G.; validation, Z.G., M.L. and B.Z.; formal analysis, Z.G.; data curation, Z.G.; writing—original draft preparation, Z.G.; writing—review and editing, Z.Z., M.L., B.Z. and T.W.; visualization, Z.G.; supervision, Z.Z. and T.W.; funding acquisition, T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No. 62003359).

Institutional Review Board Statement: Not applicable. This study used numerical simulation data and publicly available fixed-wing UAV telemetry data and did not involve human participants or animals.

Informed Consent Statement: Not applicable.

Data Availability Statement: The simulation data analyzed in this study were generated using JSBSim. The external real-flight telemetry used for the zero-shot evaluation was obtained from the publicly available IDF-DS fixed-wing UAV dataset, available through the Zenodo repository at <https://doi.org/10.5281/zenodo.16992975> and described in the corresponding Scientific Data publication cited in this manuscript. The data-generation scripts, episode splits, model configurations, trained-model information, and evaluation outputs produced in this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned aerial vehicle
U-STAR-PIML	Uncertainty-aware staged trust-adaptive residual physics-informed machine learning
PIML	Physics-informed machine learning
PINN	Physics-informed neural network
6DoF	Six degrees of freedom
GRU	Gated recurrent unit
TCN	Temporal convolutional network
FNO	Fourier neural operator
ID	In-distribution
OOD	Out-of-distribution
NED	North-east-down
RMSE	Root-mean-square error
NLL	Negative log-likelihood
PICP	Prediction interval coverage probability
MPIW	Mean prediction interval width
GPU	Graphics processing unit

Appendix A. Supplementary Methodological Details

Appendix A.1. Compact 6DoF Prior

For completeness, the compact prior uses body-frame velocity and angular-rate vectors

$$\mathbf{v}_b = [u, v, w]^T, \quad \boldsymbol{\omega}_b = [p, q, r]^T. \quad (\text{A1})$$

The standard translational and rotational rigid-body relations are

$$\dot{\mathbf{v}}_b = m^{-1} \mathbf{F}_b - \boldsymbol{\omega}_b \times \mathbf{v}_b + \mathbf{g}_b, \quad (\text{A2})$$

and

$$\dot{\boldsymbol{\omega}}_b = \mathbf{J}^{-1} [\mathbf{M}_b - \boldsymbol{\omega}_b \times (\mathbf{J} \boldsymbol{\omega}_b)]. \quad (\text{A3})$$

The attitude relation is

$$\dot{\boldsymbol{\eta}} = \mathbf{T}(\phi, \theta) \boldsymbol{\omega}_b, \quad \boldsymbol{\eta} = [\phi, \theta, \psi]^T, \quad (\text{A4})$$

and the position relation is given in Equation (9). Here, m , $\mathbf{J}$, $\mathbf{F}_b$, $\mathbf{M}_b$, and $\mathbf{g}_b$ denote mass, inertia, body-frame force, body-frame moment, and gravity, respectively. The learnable scaling coefficients in the main-text physics map are global parameters shared across episodes.

Appendix A.2. Auxiliary Loss Definitions

The normalized state and derivative losses used in Table 2 are

$$\mathcal{L}_{\text{state}} = \|S_x(\boldsymbol{\mu}_{t+\Delta t}) - S_x(\mathbf{x}_{t+\Delta t})\|_2^2, \quad (\text{A5})$$

and

$$\mathcal{L}_{\text{der}} = \|S_{\dot{x}}(\dot{\mathbf{x}}_t^{\text{fused}}) - S_{\dot{x}}(\dot{\mathbf{x}}_t)\|_2^2. \quad (\text{A6})$$

The fast-state subset and its derivative loss are

$$\mathcal{F} = \{v, w, p, q, r\}, \quad \mathcal{L}_{\text{fast}} = \left\| S_{\dot{x}, \mathcal{F}}(\dot{\mathbf{x}}_{t, \mathcal{F}}^{\text{fused}}) - S_{\dot{x}, \mathcal{F}}(\dot{\mathbf{x}}_{t, \mathcal{F}}) \right\|_2^2. \quad (\text{A7})$$

The fast-slope loss uses $\mathcal{H} = \{v, p, q, r\}$ and is

$$\mathcal{L}_{\text{slope}} = \left\| S_{\dot{x}, \mathcal{H}}\left(\frac{\boldsymbol{\mu}_{t+\Delta t, \mathcal{H}} - \mathbf{x}_{t, \mathcal{H}}}{\Delta t}\right) - S_{\dot{x}, \mathcal{H}}\left(\frac{\mathbf{x}_{t+\Delta t, \mathcal{H}} - \mathbf{x}_{t, \mathcal{H}}}{\Delta t}\right) \right\|_2^2. \quad (\text{A8})$$

The kinematic and navigation checks are

$$\mathcal{L}_{\text{kin}} = \left\| S_{\dot{x}, \text{kin}}(\dot{\mathbf{x}}_{t, \text{kin}}^{\text{fused}}) - S_{\dot{x}, \text{kin}}(h_{\text{kin}}(\mathbf{x}_t)) \right\|_2^2, \quad (\text{A9})$$

and

$$\mathcal{L}_{\text{nav}} = \left\| S_p\left(\frac{\boldsymbol{\mu}_{t+\Delta t, p} - \mathbf{x}_{t, p}}{\Delta t}\right) - S_p(\dot{\mathbf{x}}_{t, p}) \right\|_2^2. \quad (\text{A10})$$

Because hard kinematics set the corresponding fused derivative directly, the first term is identically zero and the second is close to zero for the clean reconstructed reference.

The high-frequency increment loss is

$$\mathcal{L}_{\text{freq}} = \frac{1}{|\mathcal{H}|} \left\langle \left\| \frac{\Delta \mathbf{x}_{t, \mathcal{H}}^{\text{pred}} - \Delta \mathbf{x}_{t, \mathcal{H}}^{\text{ref}}}{\mathbf{s}_{\mathcal{H}}^{\Delta}} \right\|_2^2 \right\rangle, \quad (\text{A11})$$

where $\Delta \mathbf{x}_t^{\text{pred}} = \boldsymbol{\mu}_{t+\Delta t} - \mathbf{x}_t$ and $\langle \cdot \rangle$ denotes the masked batch mean. The maneuver-weighted horizontal position-slope term is

$$\mathcal{L}_{\text{slope}}^{\text{pos}} = \left\langle (1 + \beta_s m_t) \left(\frac{\dot{p}_t^{\text{pred}} - \dot{p}_t^{\text{ref}}}{s_p} \right)^2 \right\rangle, \quad \beta_s = 2.5. \quad (\text{A12})$$

The turning-point score is

$$c_t = \text{clamp} \left[\frac{1}{2} \sigma \left(-\frac{\dot{p}_{t-1}^{\text{ref}} \dot{p}_t^{\text{ref}}}{s_p^2} \right) + \frac{1}{2} \exp \left(-\left(\frac{\dot{p}_t^{\text{ref}}}{s_p} \right)^2 \right), 0, 1 \right], \quad (\text{A13})$$

with turning-point weight $1 + \beta_t c_t$ and $\beta_t = 4.0$. The position-band term compares reconstructed horizontal positions over 41-step, 11-step, and 1-step windows with relative weights 1.0, 1.5, and 0.8, respectively. The complete position penalty is their weighted sum, evaluated only where recursive targets are available. Gate smoothness penalizes temporal differences in α_t , and the entropy regularizer penalizes near-saturated activations; both are auxiliary regularizers without a convex-mixture constraint.

References

1. Le Clainche, S.; Ferrer, E.; Gibson, S.; Cross, E.; Parente, A.; Vinuesa, R. Improving Aircraft Performance Using Machine Learning: A Review. *Aerosp. Sci. Technol.* **2023**, *138*, 108354. [CrossRef]
2. Stachiw, T.; Crain, A.; Ricciardi, J. A Physics-Based Neural Network for Flight Dynamics Modelling and Simulation. *Adv. Model. Simul. Eng. Sci.* **2022**, *9*, 13. [CrossRef]
3. Zhang, B.; Lin, X.; Zhu, Y.; Tian, J.; Zhu, Z. Enhancing Multi-UAV Reconnaissance and Search Through Double Critic DDPG with Belief Probability Maps. *IEEE Trans. Intell. Veh.* **2024**, *9*, 3827–3842. [CrossRef]
4. Ruan, B.; Tian, J.; Huang, M.; Ning, X.; Wang, J.; Zhang, B.; Zhi, F. Objective Programming Partitions and Rule-Based Spanning Tree for UAV Swarm Regional Coverage Path Planning. *Drones* **2026**, *10*, 60. [CrossRef]

5. Yu, R.; Wang, R. Learning Dynamical Systems from Data: An Introduction to Physics-Guided Deep Learning. *Proc. Natl. Acad. Sci. USA* **2024**, *121*, e2311808121. [[CrossRef](#)] [[PubMed](#)]
6. Karniadakis, G.E.; Kevrekidis, I.G.; Lu, L.; Perdikaris, P.; Wang, S.; Yang, L. Physics-Informed Machine Learning. *Nat. Rev. Phys.* **2021**, *3*, 422–440. [[CrossRef](#)]
7. Willard, J.; Jia, X.; Xu, S.; Steinbach, M.; Kumar, V. Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems. *ACM Comput. Surv.* **2022**, *55*, 1–37. [[CrossRef](#)]
8. Yin, Y.; Le Guen, V.; Doná, J.; de Bézenac, E.; Ayed, I.; Thome, N.; Gallinari, P. Augmenting Physical Models with Deep Networks for Complex Dynamics Forecasting. *J. Stat. Mech. Theory Exp.* **2021**, *2021*, 124012. [[CrossRef](#)]
9. Wang, S.; Yu, X.; Perdikaris, P. When and Why PINNs Fail to Train: A Neural Tangent Kernel Perspective. *J. Comput. Phys.* **2022**, *449*, 110768. [[CrossRef](#)]
10. Rathore, P.; Lei, W.; Frangella, Z.; Lu, L.; Udell, M. Challenges in Training PINNs: A Loss Landscape Perspective. In *Proceedings of the 41st International Conference on Machine Learning*; PMLR: New York, NY, USA, 2024; Volume 235, pp. 42159–42191.
11. Psaros, A.F.; Meng, X.; Zou, Z.; Guo, L.; Karniadakis, G.E. Uncertainty Quantification in Scientific Machine Learning: Methods, Metrics, and Comparisons. *J. Comput. Phys.* **2023**, *477*, 111902. [[CrossRef](#)]
12. Ovadia, Y.; Fertig, E.; Ren, J.; Nado, Z.; Sculley, D.; Nowozin, S.; Dillon, J.; Lakshminarayanan, B.; Snoek, J. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty under Dataset Shift. In *Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32, pp. 13991–14002.
13. Løw-Hansen, B.; Hann, R.; Gryte, K.; Johansen, T.A.; Deiler, C. Modeling and Identification of a Small Fixed-Wing UAV Using Estimated Aerodynamic Angles. *CEAS Aeronaut. J.* **2025**, *16*, 501–523. [[CrossRef](#)]
14. Benyamen, H.; Chowdhury, M.; Keshmiri, S.S. Data-Driven Aircraft Modeling for Robust RL Control Synthesis with Flight Test Validation. *J. Dyn. Syst. Meas. Control* **2024**, *146*, 061105. [[CrossRef](#)]
15. Bauersfeld, L.; Kaufmann, E.; Foehn, P.; Sun, S.; Scaramuzza, D. NeuroBEM: Hybrid Aerodynamic Quadrotor Model. In *Proceedings of the Robotics: Science and Systems XVII, Virtual*, 12–16 July 2021. [[CrossRef](#)]
16. Saviolo, A.; Li, G.; Loianno, G. Physics-Inspired Temporal Learning of Quadrotor Dynamics for Accurate Model Predictive Trajectory Tracking. *IEEE Robot. Autom. Lett.* **2022**, *7*, 10256–10263. [[CrossRef](#)]
17. Gu, W.; Primates, S.; Rizzo, A. Physics-Informed Neural Network for Quadrotor Dynamical Modeling. *Robot. Auton. Syst.* **2024**, *171*, 104569. [[CrossRef](#)]
18. Bianchi, D.; Epicoco, N.; Di Ferdinando, M.; Di Gennaro, S.; Pepe, P. Physics-Informed Neural Networks for Unmanned Aerial Vehicle System Estimation. *Drones* **2024**, *8*, 716. [[CrossRef](#)]
19. Michek, N.E.; Mehta, P.M.; Huebsch, W.W. Flight Dynamics Modeling Using Physics-Informed Neural Networks. *AIAA J.* **2025**. [[CrossRef](#)]
20. Michek, N.E.; Mehta, P.M.; Huebsch, W.W. Flight Dynamic Uncertainty Quantification Modeling Using Physics-Informed Neural Networks. *AIAA J.* **2024**, *62*, 4234–4246. [[CrossRef](#)]
21. Meng, C.; Griesemer, S.; Cao, D.; Seo, S.; Liu, Y. When Physics Meets Machine Learning: A Survey of Physics-Informed Machine Learning. *Mach. Learn. Comput. Sci. Eng.* **2025**, *1*, 20. [[CrossRef](#)]
22. Krishnapriyan, A.; Gholami, A.; Zhe, S.; Kirby, R.; Mahoney, M.W. Characterizing Possible Failure Modes in Physics-Informed Neural Networks. In *Proceedings of the Advances in Neural Information Processing Systems, Virtual*, 6–14 December 2021; Curran Associates, Inc.: Red Hook, NY, USA, 2021; Volume 34, pp. 26548–26560.
23. Levine, M.E.; Stuart, A.M. A Framework for Machine Learning of Model Error in Dynamical Systems. *Commun. Am. Math. Soc.* **2022**, *2*, 283–344. [[CrossRef](#)]
24. O'Connell, M.; Shi, G.; Shi, X.; Azizzadenesheli, K.; Anandkumar, A.; Yue, Y.; Chung, S.J. Neural-Fly Enables Rapid Learning for Agile Flight in Strong Winds. *Sci. Robot.* **2022**, *7*, eabm6597. [[CrossRef](#)] [[PubMed](#)]
25. Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A.; Anandkumar, A. Fourier Neural Operator for Parametric Partial Differential Equations. In *Proceedings of the International Conference on Learning Representations, Virtual*, 3–7 May 2021.
26. Kovachki, N.; Li, Z.; Liu, B.; Azizzadenesheli, K.; Bhattacharya, K.; Stuart, A.; Anandkumar, A. Neural Operator: Learning Maps between Function Spaces with Applications to PDEs. *J. Mach. Learn. Res.* **2023**, *24*, 1–97.
27. Abdar, M.; Pourpanah, F.; Hussain, S.; Rezazadegan, D.; Liu, L.; Ghavamzadeh, M.; Fieguth, P.; Cao, X.; Khosravi, A.; Acharya, U.R.; et al. A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges. *Inf. Fusion* **2021**, *76*, 243–297. [[CrossRef](#)]
28. Angelopoulos, A.N.; Bates, S. Conformal Prediction: A Gentle Introduction. *Found. Trends Mach. Learn.* **2023**, *16*, 494–591. [[CrossRef](#)]
29. Seitzer, M.; Tavakoli, A.; Antic, D.; Martius, G. On the Pitfalls of Heteroscedastic Uncertainty Estimation with Probabilistic Neural Networks. In *Proceedings of the International Conference on Learning Representations, Virtual*, 25–29 April 2022.

30. Xu, C.; Xie, Y. Conformal Prediction Interval for Dynamic Time-Series. In *Proceedings of the 38th International Conference on Machine Learning*; PMLR: New York, NY, USA, 2021; Volume 139, pp. 11559–11569.
31. Yang, J.; Zhou, K.; Li, Y.; Liu, Z. Generalized Out-of-Distribution Detection: A Survey. *Int. J. Comput. Vis.* **2024**, *132*, 5635–5662. [[CrossRef](#)]
32. Ho, J.; Jain, A.N.; Abbeel, P. Denoising Diffusion Probabilistic Models. In *Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–12 December 2020*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 6840–6851. [[CrossRef](#)]
33. Song, Y.; Sohl-Dickstein, J.; Kingma, D.P.; Kumar, A.; Ermon, S.; Poole, B. Score-Based Generative Modeling through Stochastic Differential Equations. In *Proceedings of the International Conference on Learning Representations, Virtual, 3–7 May 2021*.
34. Berndt, J.S. JSBSim: An Open Source Flight Dynamics Model in C++. In *Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit, Providence, RI, USA, 16–19 August 2004*; AIAA Paper 2004-4923. [[CrossRef](#)]
35. Muratore, F.; Ramos, F.; Turk, G.; Yu, W.; Gienger, M.; Peters, J. Robot Learning From Randomized Simulations: A Review. *Front. Robot. AI* **2022**, *9*, 799893. [[CrossRef](#)] [[PubMed](#)]
36. Linka, K.; Schäfer, A.; Meng, X.; Zou, Z.; Karniadakis, G.E.; Kuhl, E. Bayesian Physics Informed Neural Networks for Real-World Nonlinear Dynamical Systems. *Comput. Methods Appl. Mech. Eng.* **2022**, *402*, 115346. [[CrossRef](#)]
37. Costa, E.A.; Rebello, C.M.; Santana, V.V.; Nogueira, I.B.R. Physics-Informed Neural Network Uncertainty Assessment through Bayesian Inference. *IFAC-PapersOnLine* **2024**, *58*, 652–657. [[CrossRef](#)]
38. Riboldi, C.E.D.; Tomasoni, M. Formation Flight of Fixed-Wing UAVs: Dynamic Modeling, Guidance Design, and Testing in Realistic Scenarios. *Aerospace* **2025**, *12*, 260. [[CrossRef](#)]
39. García-Gascón, C.; Bas-Bolufer, J.; Castelló-Pedrero, P.; García-Manrique, J.A. An Open Benchmark Dataset for Machine Learning and Intelligent Trajectory Optimization in Fixed-Wing Unmanned Aerial Systems. *Sci. Data* **2026**, *13*, 364. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.