Towards a Physics-Aware Autonomous Rendezvous Transformer

Tithira Perera
Department of Computer Science and Engineering
University of Moratuwa
Katubedda, Sri Lanka
tithira.21@cse.mrt.ac.lk

Uthayasanker Thayasivam
Department of Computer Science and Engineering
University of Moratuwa
Katubedda, Sri Lanka
rtuthaya@cse.mrt.ac.lk

***Abstract*—The Autonomous Rendezvous Transformer (ART) is a Transformer-based architecture for generating spacecraft rendezvous trajectories, designed to warm-start a sequential convex optimizer for guaranteed constraint satisfaction. In this report, we describe the original ART architecture and investigate an extension to make it physics-aware by incorporating knowledge of orbital dynamics into the training process. We review related literature on physics-informed machine learning and detail our methodology of augmenting the training loss with a physics-based term that penalizes deviations from true dynamics. Finally, we present preliminary experimental results on an orbital rendezvous dataset, showing that the physics-aware ART achieves slight improvements in trajectory accuracy and optimization per formance compared to the baseline, demonstrating the promise of this approach.**

The project codebase can be accessed from https://github.com/Tithira-Perera-210460V/PA-ART.git

***Index Terms*—Autonomous Rendezvous Transformer, ART, Physics-aware machine learning, optimization**

I. Introduction

Autonomous trajectory planning for spacecraft rendezvous and docking is a challenging problem due to nonconvex dynamics and strict safety constraints. Traditional solutions rely on optimal control solvers that require a good initial guess to converge to a feasible, low-fuel trajectory. Recently, learning based approaches have been proposed to aid trajectory optimization by providing informed initial guesses quickly. In particular, the Autonomous Rendezvous Transformer (ART) introduced by Guffanti et al. [1] uses a Transformer model to generate an initial trajectory for a given rendezvous scenario, which is then refined by a Sequential Convex Programming (SCP) solver to enforce all physical constraints. This two stage framework exploits the strengths of data-driven learning for speed and coverage, while retaining the guarantees of a traditional optimizer for safety and feasibility. The original ART approach demonstrated that modern sequence models can learn near-optimal policies from historical trajectories and substantially improve the efficiency of the trajectory optimization process [1] [2]. However, the learned model in the first stage is trained purely in a data driven manner (supervised on optimal trajectory examples) without explicit incorporation of known physics beyond what is implicitly contained in the training data. This can lead to the model sometimes generating trajectories that, while statistically plausible, may violate dynamics or constraint requirements until corrected by the second-stage optimizer. Such corrections impose extra burden on the SCP solver. In this work, we explore making the ART physics-aware in order to improve the physical realism and feasibility of its generated trajectories. We focus on a simple yet effective approach: modifying the training objective of the Transformer to include a physics-based loss term. By leveraging knowledge of the spacecraft dynamics during training, we aim to bias the model toward outputs that are consistent with physical laws. This idea is inspired by the broader trend of physics-informed machine learning, which has shown that embedding physical constraints into model training can improve generalization and reliability in scientific domains.

II. Background

A. Transformers for Sequence Prediction

The Transformer architecture [3] has achieved strong results across domains including natural language processing (NLP) [4], computer vision [5], and robotics [6]. Its core appeal for sequence modeling lies in processing entire sequences in parallel, avoiding the strictly stepwise computation of recurrent networks and thereby enabling efficient learning of long-range dependencies.

At a high level, a Transformer network composes one or more *Transformer blocks* that implement a sequence-to-sequence, dimensionality-preserving map $f : \mathbb{R}^{d\times N} \to \mathbb{R}^{d\times N}$ on a length-N sequence of d-dimensional embeddings $\big(z(t_1), \ldots, z(t_N)\big)$. The core operation is *self-attention*: each input embedding $z(t_i)$ is linearly projected to a query $q_i \in \mathbb{R}^d$, key $k_i \in \mathbb{R}^d$, and value $v_i \in \mathbb{R}^d$; the layer output at position i is a similarity-weighted sum of values,

$$z(t_i) \ = \ \sum_{j=1}^{N} \mathrm{softmax}\Big(\{\langle q_i, k_{j'}\rangle\}_{j'=1}^{N}\Big)_j \, v_j, \tag{1}$$

where $\langle\cdot,\cdot\rangle$ denotes the dot product and the softmax yields nonnegative weights that sum to one. Intuitively, this lets the model "attend" to positions with large query–key similarity.

For autoregressive sequence prediction, this work adopts the GPT-style *causal* Transformer [7], which applies a causal self-attention mask so that each position i only attends to positions $j \leq i$, thereby respecting temporal order during training and inference. Concretely, the summation and softmax in (1) are restricted to the prefix $\{1, \ldots, i\}$.

B. Physics-Aware Machine Learning

Physics-aware machine learning (often termed **physics-informed machine learning**, PIML) augments data-driven models with physical principles to produce predictions that respect known laws of physics. Instead of relying purely on statistical patterns, these models embed domain-specific equations or constraints (e.g. conservation laws, kinematic equations) into the learning process. The result is typically improved data efficiency, better generalization, and physically consistent predictions. This review surveys on *Physics-Informed Neural Networks (PINN)* and *Transformer-based models*, two of the key approaches in this direction.

1) Physics-Informed Neural Networks (PINN): PINNs marked a paradigm shift by directly encoding governing equations into a neural network's training objective [8]. A PINN typically uses a deep neural network to represent the solution of a physical system and augments the data-driven loss with a physics-based term - for example, the residual of a partial differential equation (PDE) evaluated via automatic differentiation. This approach ensures physical consistency by penalizing violation of known laws (conservation equations, boundary conditions, etc.) during training. A crucial practical challenge is balancing the data fidelity loss and the physics loss. Early PINNs used manually tuned static weights for different loss terms, but recent works have introduced adaptive loss weighting strategies to improve training robustness [9]. For example, Xiang et al. (2021) [10] treated PINN training as a multi-task learning problem and used a self-adaptive weighting based on maximizing likelihood, dynamically adjusting each term's weight. Similarly, dynamic weighting methods have been proposed - Li and Feng (2022) [11] employed a minimax strategy that treats loss weights as trainable variables, automatically emphasizing regions or equations with higher error. Such schemes have been shown to accelerate convergence on stiff or multi-scale PDE problems by preventing any single loss component from dominating.

Another focus in PINNs research is ensuring stable multi-step time integration for long-duration dynamics. Standard PINNs often attempt to learn the entire spatio-temporal solution in one global model which can struggle with long-term accuracy due to compounding errors or difficulty capturing late-time behaviors. To address this, researchers have developed *time-marching PINN frameworks* that break a long simulation into sequential segments. In such approaches, the PINN is trained in a shorter time window, then its prediction at the end of the window provides initial conditions for the next segment, where a new network (or a fine-tuned network) takes over. This sequential training strategy - first demonstrated by Wight Zhao [12] and extended in later works - avoids the shortcomings of global-in-time training, such as poor utilization of local temporal features and error accumulation. It effectively ensures continuity and physical consistency across time segments, significantly improving overall accuracy and stability in solving time-evolution PDEs. For example, Chen et al. (2024) [13] introduced AT-PINN, an advanced time-marching PINN for structural vibration analysis that solves successive time segments with dedicated sub-networks, achieving stable long duration predictions where a single network would diverge.

In summary, PINNs incorporate physics via their loss functions and have evolved with strategies like adaptive loss weighting and sequential domain decomposition to better handle multi-fidelity data and long-term simulations. Applications span fluid flows [14], turbulence modeling [15], material mechanics [16], and biomedical systems [17], where PINNs often outperform purely data-driven models by respecting physical laws even with limited data.

2) Transformer-Based Models for Physical Dynamics: Transformer architectures [3], known for their sequence modeling prowess, have recently been adapted to physics-informed learning, often overlapping with the neural operator approach. Transformers can leverage global attention to capture long-range dependencies in physical systems (e.g. interactions over large spatial extents or long times) which is advantageous in multi-scale problems. A prominent example is the PDE-Transformer, a versatile transformer trained on a broad set of PDE simulations [18]. PDE-Transformer was designed as a foundation model for physics, using a multi-scale attention mechanism and treating space-time patches as tokens, enabling it to learn dynamics of different equations and resolutions in a unified framework. While primarily a data-driven model, it incorporates deep conditioning on known physical parameters (boundary conditions, PDE coefficients) by concatenating them as input embeddings, ensuring that physical context steers the attention layers. This conditioning acts analogous to a soft physics constraint: the transformer learns to adjust its predictions based on the provided physics information, thus maintaining consistency with, for example, changed boundary conditions or source terms. Transformers have also been combined with U-Nets or CNN backbones to enforce locality (for fine details) alongside global attention. In one approach, a U-Net Transformer model was made physics-informed by injecting PDE constraints at various layers and training it to extrapolate far beyond its seen time horizon [19]. Liang et al. [20] demonstrated a model that, given only a short initial trajectory of a dynamical system, could predict arbitrarily long futures by virtue of an architecture respecting the system's differential equations. This "beyond training horizon" capability indicates that the network effectively learned the underlying physical rules, not just a short-term mapping.

Overall, the integration of physics into machine learning frameworks has significantly enhanced the reliability and interpretability of predictive models. While *Physics-Informed Neural Networks* achieve this through explicit incorporation of governing equations into their loss functions, *Transformer-based architectures* embed physical consistency via global attention and parameter conditioning across space-time domains. Both paradigms aim to balance data fidelity with adherence to physical laws - often through adaptive weighting schemes and multi-step rollout strategies - to ensure stable, long-horizon predictions.

III. Autonomous Rendezvous Transformer (ART) [1]

A. Overview and Problem Setting

ART is a Transformer-based framework for long-horizon trajectory generation in constrained optimal control problems (OCPs), designed to both imitate near-optimal policies and *warm-start* a sequential optimizer so that hard constraints are ultimately enforced by optimization. In contrast to purely end-to-end learning or fixed-parameter trajectory parameterizations, ART leverages sequence modeling to generate time-dependent trajectories and then hands them to a Sequential Convex Programming (SCP) solver for guaranteed constraint satisfaction.

The study considers a generic time-discretized, non-convex, constrained OCP with state $x \in \mathbb{R}^s$, control $u \in \mathbb{R}^a$, nonlinear dynamics $f : \mathbb{R}^{s+a} \to \mathbb{R}^s$, generic state and action constraint set $\mathcal{C}$, and horizon length $N \in \mathbb{N}$:

$$\begin{aligned} \min_{\{x(t_i),u(t_i)\}_{i=1}^N} \quad & \sum_{i=1}^{N} J\big(x(t_i), u(t_i)\big) \\ \text{s.t.} \quad & x(t_{i+1}) = f\big(x(t_i), u(t_i)\big), \\ & x(t_i),\, u(t_i) \in \mathcal{C}(t_i). \end{aligned} \tag{2}$$

ART introduces a *performance-conditioned* sequence representation that prepends, at every time step, two scalars: the reward-to-go $R(t_i)$ (negative accumulated cost-to-go) and the constraint-to-go $C(t_i)$ (accumulated future constraint violations). Concretely,

$$\begin{aligned} \tau = \big(P(t_1), x(t_1), u(t_1),\, \dots,\, P(t_N), x(t_N), u(t_N)\big), \\ \text{where } P = \{R, C\}. \end{aligned} \tag{3}$$

$$R(t_i) = \sum_{j=i}^{N} R\big(x(t_j), u(t_j)\big) = -\sum_{j=i}^{N} J\big(x(t_j), u(t_j)\big) \tag{4}$$

and

$$\begin{aligned} C(t_i) &= \sum_{j=i}^{N} C(t_j) \\ & where \\ C(t_j) &= \begin{cases} 1, & \exists\, x(t_j), u(t_j) \notin \mathcal{C}(t_j), \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{5}$$

so that trajectories can be generated *conditional* on a user-specified $(R(t_1), C(t_1))$ at inference.

B. Architecture

Given a maximum context length K, ART ingests the last K time steps of the four modalities (reward-to-go, constraint-to-go, state, control), for a total of $4K$ tokens. Each modality is linearly projected to the model dimension via a modality-specific encoder; learned time-step embeddings are added; and a GPT-style causal Transformer processes the sequence to autoregressively predict future states and controls.

At the self-attention core, each embedding is mapped to queries, keys, and values and updated by a softmax-weighted similarity over the (causally masked) prefix, enabling information flow along the sequence without violating temporal order.

a) Reference hyperparameters: The implementation (PyTorch + HuggingFace) uses 6 layers, 6 attention heads, 384-d embeddings, context length $K{=}100$, ReLU nonlinearity, dropout 0.1, AdamW with learning rate 3×10^{-5}, gradient norm clipping 1.0, batch size 4 with gradient accumulation of 8 steps.

C. Training Objective

Trajectories solved from (2) are reorganized to match (3) and used for teacher-forced sequence modeling. Denoting ART parameters by θ, the loss sums squared prediction errors for state and control at each step:

$$\mathcal{L}(\tau) = \sum_{n=1}^{N_d} \sum_{i=1}^{N} \left(\|x_n(t_i) - \hat{x}_n(t_i)\|_2^2 + \|u_n(t_i) - \hat{u}_n(t_i)\|_2^2 \right)$$

where

$$\begin{aligned} \hat{x}_n(t_i) &= \mathrm{ART}_\theta\big(\tau_n^{<t_i}\big)\,, \\ \hat{u}_n(t_i) &= \mathrm{ART}_\theta\big(\tau_n^{<t_i}, x_n(t_i)\big)\,. \end{aligned} \tag{6}$$

Here $\tau^{<t_i}$ is the prefix up to t_{i-1}.

D. Inference Pipelines

Once trained, ART can predict actions (controller), states (learned dynamics), and even performance parameters. Two pipelines are evaluated:

(i) Transformer-only: states are predicted by the Transformer itself.

(ii) Dynamics-in-the-loop: ART predicts actions but state rollouts are computed with the known dynamics f.

In both schemes, after executing $u(t_i)$ for $x(t_i)$, the algorithm updates $R(t_{i+1}) = R(t_i) - (-J(x(t_i), u(t_i)))$ and $C(t_{i+1}) = C(t_i) - C(t_i)$, then proceeds.

E. Application to Spacecraft Rendezvous

For rendezvous with a target on orbit $\boldsymbol{\varpi} \in \mathbb{R}^6$, the relative motion is modeled in either RTN Cartesian coordinates $\delta\chi = \{\delta r, \delta v\} \in \mathbb{R}^6$ or Relative Orbital Elements (ROE) $\delta\boldsymbol{\varpi} = \{\delta a, \delta\lambda, \delta e_x, \delta e_y, \delta i_x, \delta i_y\} \in \mathbb{R}^6$, with a first-order map $\delta\chi(t) \approx \Psi(t)\,\delta\boldsymbol{\varpi}(t)$. Time-discretized, perturbed relative dynamics on ROE take the form

$$\delta\boldsymbol{\varpi}(t_{i+1}) \;=\; \Phi(\Delta t, t_i)\,\delta\boldsymbol{\varpi}(t_i) \;+\; \Phi(\Delta t, t_i) B(t_i)\,\Delta v(t_i), \tag{7}$$

where Φ captures relevant perturbations and B is the control input matrix.

Three OCPs are studied: (1) a convex two-point boundary value problem (TPBVP), (2) a convex rendezvous problem with a pre-docking waypoint and approach-cone (SOC) constraint, and (3) a non-convex rendezvous problem additionally enforcing a keep-out-zone (KOZ) ellipsoid around the target; the non-convex KOZ is sequentially linearized within SCP and regulated by a second-order-cone trust region with an exponential shrink policy and standard cost/stopping tests.

F. Experimental Setup

A large offline dataset is generated for an ISS-like scenario: randomized, *passively safe* initial ROE outside the KOZ; randomized horizons; and both non-convex (Problem 3) solutions and their convex-relaxation warm-starts. The dataset contains $400{,}000$ trajectories with a 90/10 train/test split.

G. Results

a) Forecasting (Imitation): Across convex and non-convex datasets, ART consistently outperforms GRU and LSTM baselines on state and control metrics, especially when rolling out with dynamics-in-the-loop; using known dynamics substantially improves control-profile imitation (e.g., impulse accuracy and control RMSE). Moreover, ART can reliably *track* desired performance-conditioning $(R(t_1), C(t_1))$ within the range supported by the data.

b) Control (Warm-starting SCP): When used to warm-start the SCP for the non-convex rendezvous problem, ART is compared to warm-starts from (i) TPBVP and (ii) convex rendezvous without KOZ; ART is initialized with $R(t_1)$ equal to the negative total cost of the convex warm-start and $C(t_1) = 0$. ART warm-starts reduce suboptimality relative to convex baselines (often $> 20\%$ average improvement when the convex solution initially violates KOZ), cut SCP iterations by up to ≈ 1.5 on difficult instances, and yield similar total runtime despite higher warm-start computation time (ART ~ 0.5s vs convex ~ 0.15s), because SCP converges faster from ART's initializations. Unfeasibility rates are comparable to the convex warm-start and markedly better than TPBVP.

H. Discussion and Takeaways

ART provides (i) high-fidelity sequence modeling of optimal trajectories, (ii) controllable generation via performance-conditioning, and (iii) effective warm-starts that preserve the guarantees of optimization while leveraging data-driven structure. Empirically, ART yields near-optimal trajectories efficiently and compares favorably to strong baselines, opening directions in robustness, uncertainty handling, and closed-loop control under contingencies.

IV. Physics-Aware Enhancement of the Autonomous Rendezvous Transformer

A. Overview

While the baseline ART [1] achieves high fidelity in imitating expert demonstrations, it is inherently data-driven and not constrained by physical laws of motion. This can result in dynamically inconsistent predictions or infeasible trajectories, especially under extrapolated orbital regimes. To address this, a physics-aware modification of ART, denoted as **Physics-Aware ART (PA-ART)**, is introduced to integrate orbital dynamics constraints into the learning process.

B. Physics-Aware Loss

The primary enhancement introduces a composite loss function that combines imitation-based reconstruction terms with physics-consistency regularization. Given the true and predicted trajectories $\{x_t, u_t\}$ and $\{\hat{x}_t, \hat{u}_t\}$ respectively, the total training loss is defined as:

$$\mathcal{L}(\tau) = \sum_{t=1}^{T} \left(\|x_t - \hat{x}_t\|_2^2 + \|u_t - \hat{u}_t\|_2^2 \right) + \lambda_{\text{dyn}} \mathcal{L}_{\text{dyn}} + \alpha_{\text{roll}} \mathcal{L}_{\text{roll}} \tag{8}$$

where the first two terms correspond to the baseline ART objective, and the additional terms impose physics-awareness:

a) ***Physics-Consistency Loss**:* The physics-consistency loss evaluates whether the control commands predicted by the Transformer yield physically valid state transitions under orbital dynamics. At each timestep t, the predicted control $\hat{u}_t$ is applied to the *true current state* x_t through the analytical dynamics model $f(\cdot)$ to obtain the next physically propagated state $x_{t+1}^{\text{phys}} = f(x_t, \hat{u}_t)$, where f represents the discrete orbital propagator based on the Hill-Clohessy-Wiltshire and J_2-perturbed relative motion equations (same as in baseline ART implementation).

The physics loss then measures the mean-squared deviation between this physically propagated state and the ground-truth next state x_{t+1}:

$$\begin{aligned} \mathcal{L}_{\text{dyn}} &= \frac{1}{T-1} \sum_{t=1}^{T-1} \left\| x_{t+1} - x_{t+1}^{\text{phys}} \right\|_2^2 \\ &= \frac{1}{T-1} \sum_{t=1}^{T-1} \left\| x_{t+1} - f(x_t, \hat{u}_t) \right\|_2^2 \end{aligned} \tag{9}$$

This formulation ensures that the model's predicted controls generate trajectories consistent with the underlying orbital mechanics, even when the model is trained primarily through imitation learning.

The total training objective includes this term scaled by a tunable coefficient λ_{dyn}:

$$\mathcal{L} = \mathcal{L}_{\text{imit}} + \lambda_{\text{dyn}} \, \mathcal{L}_{\text{dyn}}. \tag{10}$$

Following the principles of Physics-Informed Neural Networks (PINNs) [8] [21], introducing the weighting factor λ_{dyn} stabilizes optimization and controls the balance between data fidelity and physical consistency.

b) ***Rollout Consistency Loss**:* The rollout consistency loss extends the one-step dynamics constraint to a multi-step horizon, enforcing long-term physical consistency of the predicted control sequence. Starting from a randomly sampled true state x_t, the model's predicted controls

$\{\hat{u}_t, \hat{u}_{t+1}, \ldots, \hat{u}_{t+H-1}\}$ across a rollout horizon of length H are successively applied to the physical propagation model $f(\cdot)$ to generate a simulated rollout:

$$x_{t+k+1}^{\text{phys}} = f(x_{t+k}^{\text{phys}}, \hat{u}_{t+k}) \quad k = 0, 1, \ldots, H-1 \tag{11}$$

This produces a sequence of physically propagated states $\{x_{t+1}^{\text{phys}}, \ldots, x_{t+H}^{\text{phys}}\}$, which are compared against the corresponding ground-truth states from the dataset. The rollout loss is formulated as:

$$\mathcal{L}_{\text{roll}} = \sum_{k=0}^{H-1} \left\| x_{t+k+1} - x_{t+k+1}^{\text{phys}} \right\|_2^2 \tag{12}$$

where H denotes the rollout horizon length.

Unlike the one-step dynamics loss, which verifies immediate state consistency, $\mathcal{L}_{\text{roll}}$ penalizes cumulative drift over multiple time steps, thus improving long-horizon trajectory stability. This term is incorporated into the overall loss function as:

$$\mathcal{L} = \mathcal{L}_{\text{imit}} + \lambda_{\text{dyn}}\, \mathcal{L}_{\text{dyn}} + \alpha_{\text{roll}}\, \mathcal{L}_{\text{roll}}, \tag{13}$$

where α_{roll} regulates the influence of the multi-step constraint.

Incorporating multi-step physical rollouts has been shown to improve long-term prediction stability and reduce compounding errors in sequential models [22] [23]. This regularization encourages the model to produce control sequences that not only match local transitions but also generate dynamically feasible orbital trajectories over extended horizons.

The coefficients λ_{dyn} and α_{roll} act as weighting hyperparameters, gradually increased during training:

$$\begin{aligned} \lambda_{\text{dyn}} &\leftarrow \min\bigl(\lambda_{\text{dyn}} \times 10,\ \lambda_{\text{dyn}}^{\max}\bigr) \\ \alpha_{\text{roll}} &\leftarrow \min\bigl(\alpha_{\text{roll}} \times 10,\ \alpha_{\text{roll}}^{\max}\bigr) \end{aligned} \tag{14}$$

This incremental scheduling allows the Transformer to first learn the underlying data distribution and progressively emphasize adherence to physical constraints.

The inclusion of physics-consistent losses introduces an inductive bias that guides the Transformer towards generating dynamically feasible trajectories. By penalizing deviations from physically plausible state transitions, PA-ART achieves improved one-step prediction accuracy and smoother rollout stability, while maintaining the representational flexibility of the Transformer backbone.

C. Training Process

The complete training pipeline of the physics-aware ART comprises three major stages: dataset generation, preprocessing, and model training.

1) Dataset Generation: Synthetic rendezvous datasets were generated using a convex optimization-based orbital control solver that produces feasible state–action trajectories under linearized dynamics. Each trajectory $\tau_i = \{x_i(t), u_i(t)\}$ spans $T = 100$ timesteps with six-dimensional states (relative RTN position and velocity) and three-dimensional control inputs $(\Delta v_x, \Delta v_y, \Delta v_z)$. Auxiliary quantities such as reward-to-go (r_t) and constraint-to-go (c_t) were computed to encode mission-specific objectives and constraint margins.

2) Dataset Preprocessing: All datasets (training and validation) were normalized using the statistics (mean and standard deviation) computed from the *training set only*, preventing data leakage. For a given variable z, normalization was applied as:

$$z_{\text{norm}} = \frac{z - \mu_{\text{train}}}{\sigma_{\text{train}}} \tag{15}$$

3) Model Training: The PA-ART model retains the baseline Decision Transformer architecture. The physics-aware loss coefficients were initialized and progressively increased according to the schedule above. Gradients were clipped to 0.5 for numerical stability, and validation checkpoints were saved every 1000 steps.

To enhance training stability, an *Exponential Moving Average (EMA)* of the model parameters was maintained throughout training. After each optimization step, the EMA parameters were updated according to:

$$\theta_{\text{ema}} \leftarrow \text{decay}\, \theta_{\text{ema}} + (1 - \text{decay})\, \theta \tag{16}$$

where θ denotes the current model parameters and decay $\in [0, 1)$ is a smoothing coefficient (typically close to 1). This moving average effectively filters high-frequency noise from gradient updates, resulting in smoother validation curves and more consistent convergence. During validation, the averaged parameters are temporarily swapped into the model, ensuring that performance evaluation reflects the stabilized EMA weights rather than instantaneous updates. Such averaging strategies are known to improve generalization and reduce overfitting in deep sequence models and reinforcement learning frameworks [24].

Table I gives the values of the hyperparameters used in training of the PA-ART.

TABLE I
TRAINING HYPERPARAMETERS USED FOR THE PHYSICS-AWARE ART MODEL

Parameter	**Symbol / Notation**	**Value**
General Settings		
Learning rate	η	3×10^{-5}
Batch size	B	8
Number of epochs	E	5
Optimizer		AdamW
Gradient clipping	$\\|\nabla\\|_{\max}$	0.5
Physics-Aware Loss Weights		
Initial dynamics weight	λ_{dyn}	1×10^{-4}
Initial rollout weight	α_{roll}	1×10^{-5}
Maximum dynamics weight	$\lambda_{\text{dyn}}^{\max}$	1×10^{-2}
Maximum rollout weight	$\alpha_{\text{roll}}^{\max}$	1×10^{-2}
Rollout horizon	H	3
EMA Configuration		
EMA decay factor	β_{ema}	0.999
Validation interval		every 2000 steps
Early stopping patience		8 validations

V. VALIDATION AND RESULTS

Both the baseline ART and physics-aware ART were evaluated on identical validation datasets ($N = 10\,000$, $T = 100$, $D_x = 6$, $D_u = 3$). Table II summarizes the average quantitative results obtained.

TABLE II

COMPARISON OF BASELINE ART AND PHYSICS-AWARE ART ON VALIDATION DATASET. LOWER IS BETTER FOR ALL METRICS.

Model	$\mathcal{L}_{\text{total}}$	$\mathcal{L}_{\text{state}}$	$\mathcal{L}_{\text{action}}$	physics_res	MSE@10	MSE@50	Feas. Ratio
Baseline ART	6.941×10^{-1}	4.257×10^{-3}	6.899×10^{-1}	2.626×10^{-1}	7.973×10^{2}	5.169×10^{3}	0.0
Physics-Aware ART	$\mathbf{6.887 \times 10^{-1}}$	$\mathbf{3.570 \times 10^{-5}}$	$\mathbf{6.886 \times 10^{-1}}$	$\mathbf{2.531 \times 10^{-1}}$	9.070×10^{2}	$\mathbf{5.089 \times 10^{3}}$	0.0

The results show that the physics-aware enhancement yields:

- a **99% reduction** in one-step state prediction error ($\mathcal{L}_{\text{state}}$),
- a **3.6% decrease** in physics residuals, indicating improved adherence to orbital dynamics,
- a marginally lower overall validation loss ($\mathcal{L}_{\text{total}}$), and
- improved medium-horizon stability (MSE@50).

The control prediction loss ($\mathcal{L}_{\text{action}}$) shows a slight improvement of approximately 0.77%, indicating that the physics-aware regularization not only preserves but marginally enhances the model's ability to imitate optimal control actions. However, both models still exhibit numerical divergence at long horizons (MSE@100) and a zero feasibility ratio. This degradation in long-term convergence is likely attributed to the absence of Sequential Convex Programming (SCP) refinement during data generation and training, where only the optimally solved convex trajectories were utilized without iterative feasibility correction.

a) Interpretation: These results confirm that incorporating physics-informed regularization improves local dynamic consistency and mid-horizon rollout stability without compromising policy imitation performance. The approach effectively biases the Transformer toward physically valid transitions, bridging the gap between pure data-driven learning and physically grounded trajectory generation.

VI. CONCLUSION

This work presents a physics-aware enhancement of the Autonomous Rendezvous Transformer (ART) framework for spacecraft trajectory prediction and planning. By integrating analytical orbital dynamics directly into the learning process through physics-consistency and multi-step rollout losses, the proposed Physics-Aware ART (PA-ART) achieves improved dynamic fidelity while retaining the representational flexibility of the Transformer architecture.

Experimental results demonstrated that the inclusion of physics-based regularization significantly reduced the one-step state prediction error and improved mid-horizon stability, while also yielding a modest 0.77% reduction in control prediction loss. These findings indicate that enforcing physical consistency does not compromise control imitation performance - in fact, it provides additional structure that benefits the learning of optimal actions.

Nevertheless, both the baseline and physics-aware models exhibited long-horizon divergence and zero feasibility ratio, suggesting that constraint satisfaction remains a key challenge. The loss of long-term convergence is likely due to the exclusion of Sequential Convex Programming (SCP) refinement in the dataset generation process, where only the single-step convex-optimal trajectories were used without iterative feasibility correction. Future work will therefore focus on incorporating SCP-based data augmentation and physics-guided fine-tuning to further stabilize long-term rollouts and enforce constraint adherence.

Overall, the proposed PA-ART framework illustrates a promising direction for combining deep sequence models with explicit physical priors in trajectory optimization. By bridging data-driven learning and orbital mechanics, it moves toward a new generation of intelligent, physics-consistent guidance models capable of generalizing across complex space rendezvous scenarios.

REFERENCES

[1] T. Guffanti, D. Gammelli, S. D'Amico, and M. Pavone, "Transformers for trajectory optimization with application to spacecraft rendezvous," in *Proc. IEEE Aerospace Conf.*, 2024, pp. 1–9.

[2] D. Celestini, D. Gammelli, T. Guffanti, S. D'Amico, E. Capello and M. Pavone, "Transformer-Based Model Predictive Control: Trajectory Optimization via Sequence Modeling," in IEEE Robotics and Automation Letters, vol. 9, no. 11, pp. 9820-9827, Nov. 2024, doi: 10.1109/LRA.2024.3466069.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, "Attention is all you need." Advances in neural information processing systems. 2017;30.

[4] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018. [Online]. Available: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

[5] A. Dosovitskiy, L. Beyer *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in International Conference on Learning Repre sentations, 2021.

[6] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Learning humanoid locomotion with transformers," arXiv preprint arXiv:2303.03381, 2023. [Online]. Available: https://arxiv.org/abs/2303.03381

[7] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019. [Online]. Available: https://arxiv.org/abs/1706.03762

[8] M. Raissi, P. Perdikaris, and G.E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," Journal of Computational Physics, vol. 378, no. 378, pp. 686–707, Feb. 2019, doi: https://doi.org/10.1016/j.jcp.2018.10.045.

[9] M. Raissi, P. Perdikaris, N. Ahmadi, and G.E. Karniadakis, "Physics-Informed Neural Networks and Extensions," Frontiers of Science Awards 2024 for Math/CS/Phys, 2024. Available: https://arxiv.org/pdf/2408.16806v1

[10] Z. Xiang, W. Peng, X. Zheng, X. Zhao, and W. Yao, "Self-adaptive loss balanced Physics-informed neural networks for the incompressible Navier-Stokes equations," 2021. Available: https://arxiv.org/pdf/2104.06217

[11] S. Li and X. Feng, "Dynamic Weight Strategy of Physics-Informed Neural Networks for the 2D Navier–Stokes Equations," Entropy, vol. 24, no. 9, p. 1254, Sep. 2022, doi: https://doi.org/10.3390/e24091254.

[12] C.L. Wight and J. Zhao, "Solving Allen-Cahn and Cahn-Hilliard Equations using the Adaptive Physics Informed Neural Networks," arXiv.org, 2020. https://arxiv.org/abs/2007.04542

[13] Z. Chen, S.K. Lai, and Z. Yang, "AT-PINN: Advanced time-marching physics-informed neural network for structural vibration analysis," Thin-walled structures, vol. 196, pp. 111423–111423, Mar. 2024, doi: https://doi.org/10.1016/j.tws.2023.111423.

[14] B. Steinfurth and J. Weiss, "Assimilating experimental data of a mean three-dimensional separated flow using physics-informed neural networks," Physics of fluids, vol. 36, no. 1, Jan. 2024, doi: https://doi.org/10.1063/5.0183463.

[15] S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. Karniadakis, "Physics-Informed Neural Networks (PINNs) for Heat Transfer Problems," Journal of Heat Transfer, Mar. 2021, doi: https://doi.org/10.1115/1.4050542.

[16] E. Zhang, M. Dao, G.E. Karniadakis, and S. Suresh, "Analyses of internal structures and defects in materials using physics-informed neural networks," Science Advances, vol. 8, no. 7, Feb. 2022, doi: https://doi.org/10.1126/sciadv.abk0644.

[17] J.A. Rodrigues, "Using Physics-Informed Neural Networks (PINNs) for Tumor Cell Growth Modeling," Mathematics, vol. 12, no. 8, p. 1195, Apr. 2024, doi: https://doi.org/10.3390/math12081195.

[18] B. Holzschuh, Q. Liu, G. Kohl, and N. Thuerey, "PDE-Transformer: Efficient and Versatile Transformers for Physics Simulations," arXiv.org, 2025. https://arxiv.org/abs/2505.24717.

[19] Y. Liu, J. Wang, N. Zhang, X. Qian, and S. Chai, "Beyond training horizons: A physics-informed U-Net transformer for time-dependent CO2 storage simulation in environmental applications," Applied Soft Computing, vol. 185, p. 113987, Sep. 2025, doi: https://doi.org/10.1016/j.asoc.2025.113987.

[20] Y. Liang, R. Niu, J. Yue, and M. Lei, "A Physics-Informed Recurrent Neural Network for Solving Time-Dependent Partial Differential Equations," International Journal of Computational Methods, vol. 21, no. 10, May 2023, doi: https://doi.org/10.1142/s0219876223410037.

[21] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," Nature Reviews Physics, vol. 3, no. 6, pp. 422–440, Jun. 2021, doi: https://doi.org/10.1038/s42254-021-00314-5.

[22] E. Parisotto et al., "Stabilizing Transformers for Reinforcement Learning," PMLR, pp. 7487–7498, Nov. 2020, Available: https://proceedings.mlr.press/v119/parisotto20a

[23] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to Trust Your Model: Model-Based Policy Optimization." Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/5faf461eff3099671ad63c6f3f094f7f-Paper.pdf

[24] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," Neural Information Processing Systems, 2017. https://proceedings.neurips.cc/paper/2017/hash/68053af2923e00204c3ca7c6a3150cf7-Abstract.html