Stabilizing 3D Continuum-Arm Rollouts via Equilibrium Anchoring and Feature-Lifted Residual Learning

Ahsan Tanveer[1,2], Rahdar Hussain Afridi[1,2,3,†], Waqar Hussain Afridi[1,2,†], Feitian Zhang[2,4], and Guangming Xie[1,2,5,*]

[1]Intelligent Biomimetic Design Lab, School of Advanced Manufacturing and Robotics, Peking University
[2]State Key Laboratory of Turbulence and Complex Systems, Peking University [3]FAW Robotics
[4]Robotics and Control Lab, School of Advanced Manufacturing and Robotics, Peking University
[5]Institute of Ocean Research, Peking University
[†]Equal contribution [*]Correspondence to: Guangming Xie <xiegming@pku.edu.cn>
https://anchored-rollouts.github.io/

***Abstract*—Multi-step motion prediction for continuum robots is difficult, especially under actuation distribution shift, where error accumulation can distort the predicted steady response and destabilize rollouts. This paper introduces a hybrid equilibrium-anchored residual-learning framework for a tendon-driven 3D continuum arm that makes steady behavior explicit. An equilibrium prior is learned from inexpensive static equilibrium data and used in a contractive update that continuously pulls predictions toward the equilibrium estimate, improving rollout stability. A lightweight feature-lifted residual model, linear in parameters, learns the remaining one-step mismatch from dynamic trajectory data, recovering transient dynamics. The approach is validated on 200-step simulation rollouts under stronger and faster actuation than in training, with an additional soft-tail hardware test under actuation-frequency shift. The Hybrid method reduces backbone position RMSE by 26% and tip position RMSE by 27%, producing consistent accuracy gains over prior-only and residual-only predictors while remaining stable across all tested trajectories. The same proposed model also improves robustness on standard nonlinear benchmarks against a combined Koopman baseline under matched evaluation protocols.**

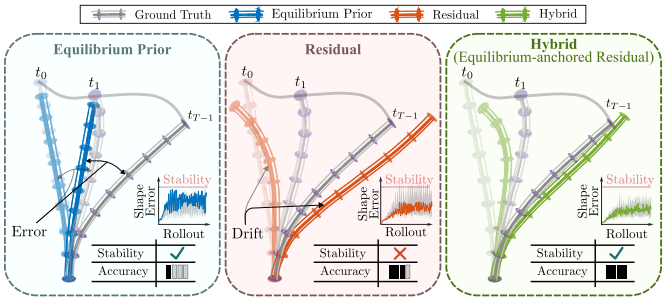

Fig. 1. **Graphical abstract.** Recursive rollout comparison for a tendon-driven continuum arm under actuation shift. Each panel overlays GROUND TRUTH (gray) and one predictor at three times (t_0, t_1, t_{T-1}): EQUILIBRIUM PRIOR (blue) remains stable but exhibits a large steady-state error, RESIDUAL (orange) captures transient dynamics but can drift, and HYBRID (green) suppresses drift while reducing overall shape error. The scorecard summarizes relative stability and shape accuracy. (Inset error plots are not to scale, and shown for visualization only.)

I. INTRODUCTION

Continuum robots are highly dexterous, but their compliance makes prediction difficult. High-dimensional nonlinear dynamics and inevitable model mismatch cause recursive rollouts to drift and occasionally become unstable [1, 2]. This is limiting for horizon-based methods such as model predictive control and sampling-based planning. Moreover, deployment conditions rarely match the training distribution exactly. Actuation profiles can differ in amplitude, frequency content, or operating regime, which further amplifies drift [3].

Prior work on multi-step prediction for continuum robots includes model-free feedback in constrained regimes and sensing-based reconstruction for closed-loop control when accurate models are hard to obtain [4, 5]. Broadly, approaches fall into physics-based modeling and data-driven learning. Physics models are interpretable and can generalize, but high-fidelity continuum-arm dynamics are difficult to capture due to distributed compliance and hard-to-identify effects such as hysteresis, friction, and viscoelasticity. Accurate simulators are also often too costly for planning or MPC, so practical models rely on simplifications that can become biased as conditions change [6, 7]. Data-driven predictors can fit complex input–output relationships and have recently seen improvements such as data-efficient learning and control via model-based learning, nonparametric methods, Neural-ODEs, and discrete-time neural solvers [8, 9, 10]. However, they often drift or become unstable under actuation shifts in amplitude, or frequency content, unless retrained or augmented with new data [11, 12, 13, 14].

A promising direction is residual hybrid modeling, which combines a coarse prior model with a learned correction. The prior provides qualitative validity and a stabilizing bias, while the correction targets systematic mismatch and unmodeled effects. For example, [15] learn residual forces to bridge simulator and real soft-robot behavior without learning full dynamics from scratch. [16] use a residual model that preserves a Koopman linear structure in lifted coordinates, which supports controller compatibility and fast evaluation while improving accuracy over an idealized prior. Related hybrids that blend analytical structure with learning have also improved performance and data efficiency in soft-robot pipelines [17]. Despite these successes, residual-physics and

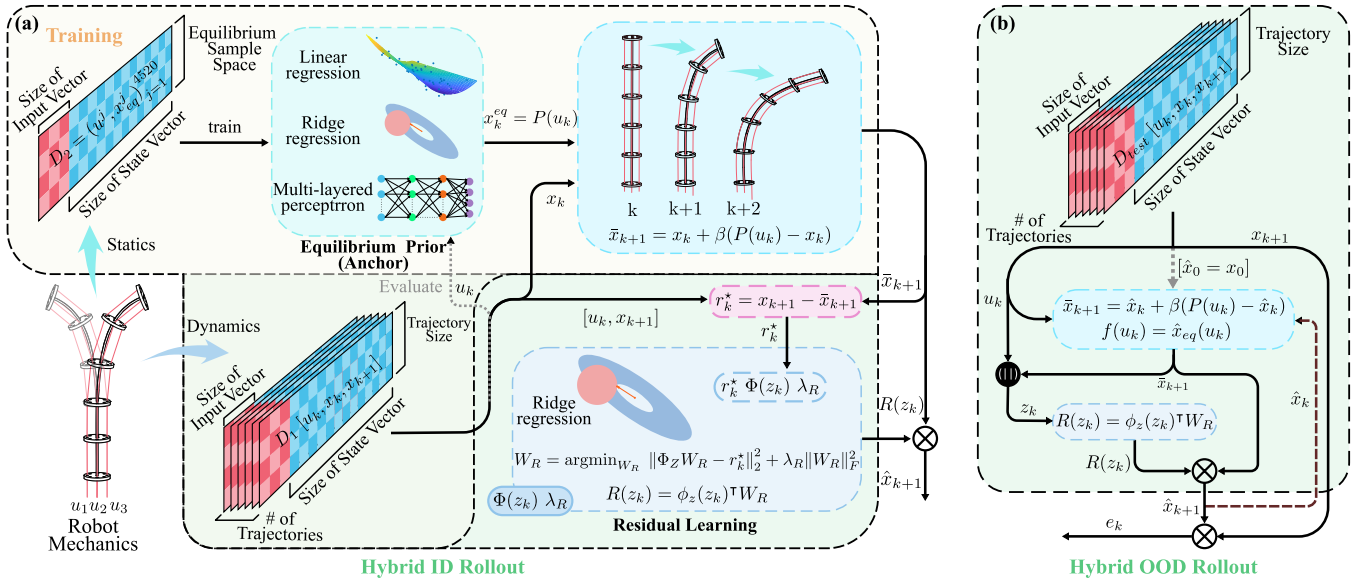

Fig. 2. **Equilibrium-anchored residual learning for continuum-arm prediction under actuation shift. (a) Training and in-distribution (ID) rollout setup.** Learn an equilibrium prior $P(\cdot)$ from the static equilibrium dataset D_2 (via linear/ridge regression or an MLP); evaluate it along D_1 to form the anchored intermediate state $\bar{x}_{k+1}$ and the prior-consistent residual target $r_k^\star = x_{k+1} - \bar{x}_{k+1}$; then fit a feature-lifted residual map $R(z) = \phi_z(z)^\top W_R$ with $z = [\bar{x}_{k+1}; u_k]$. **(b) Out-of-distribution (OOD) rollout.** Given an input sequence, iterate the anchored update and residual correction to produce multi-step predictions and compute rollout error against ground truth.

residual-Koopman models typically improve prediction by learning an additive correction to a nominal one-step dynamics update or prior operator. In contrast, our method uses a learned steady-state reference from static equilibrium pairs as a per-step rollout anchor and learns the transient deviation around that anchor. Thus, the key distinction is not simply the use of an additive residual, but the explicit separation of steady-state anchoring from transient correction to reduce drift in long recursive predictions.

In this paper, we propose an equilibrium-anchored predictor that leverages the continuum arm static input–equilibrium relationship to improve open-loop multi-step prediction under actuation shift, where recursive iteration amplifies one-step errors and exposes missing dynamics. To quantify both accuracy and robustness under operating changes, we rollout all methods under identical input sequences and report both rollout error and failure rates, since robustness under perturbations and operating changes is central in continuum and soft-robot control [18, 19]. We learn an actuation-conditioned equilibrium map from a separate equilibrium dataset and use it to anchor the prediction at each step, pulling the state toward a physically plausible steady configuration and suppressing drift. On top of this anchor, we learn a residual correction that captures transient behavior and systematic dynamic mismatch. The residual is feature-lifted and linear-in-parameters, trained on anchored states to reduce train–deploy mismatch under recursive prediction while retaining a control-friendly structure amenable to regularization and stability analysis. We study both a statics-based ridge anchor and a neural anchor, with and without residual augmentation, and evaluate on 200-step (6 s) OOD rollouts with higher-amplitude and higher-dominant-frequency tendon inputs than training. A visual summary is provided in Fig. 1. Although evaluated primarily on a tendon-driven continuum arm, the formulation applies to systems where an input-conditioned steady response can be estimated separately from transient dynamics. Our contributions are: (i) an actuation-conditioned equilibrium prior learned from static data with a contractive anchor update to reduce steady-state drift, and (ii) a prior-consistent, feature-lifted linear residual trained on anchored transitions to improve OOD multi-step accuracy.

The remainder of the paper describes the simulator and datasets, details the proposed model, and reports comparisons to data-driven and anchor-only baselines under out-of-distribution actuation, including additional nonlinear benchmarks.

II. Methodology

This section describes the proposed HYBRID predictor. Let $x_k \in \mathbb{R}^{144}$ denote the reduced state and $u_k \in \mathbb{R}^3$ the tendon command at step k, with rollouts evaluated over $T = 200$ steps ($dt = 0.03$ s). The method uses two datasets with distinct roles: an equilibrium dataset $D_2 = \{(u^{(j)}, x_{\mathrm{eq}}^{(j)})\}_{j=1}^{4520}$, where $u^{(j)} \in \mathbb{R}^3$ is a constant tendon command and $x_{\mathrm{eq}}^{(j)} \in \mathbb{R}^{144}$ is the corresponding converged static-equilibrium state, and a trajectory dataset $D_1 = \{(x_k, u_k, x_{k+1})\}$ providing dynamic transitions. From D_2 we learn an actuation-conditioned equilibrium prior $P(u_k) \approx x_k^{eq}$ that serves as a stabilizing reference, and from D_1 we learn residual correction that captures transient dynamics and systematic mismatch. Fig. 2 summarizes the training pipeline and rollout recursion.

A. Equilibrium Anchor

We first learn the physics equilibrium anchor from the static dataset D_2. Specifically, we fit an actuation-conditioned

equilibrium map $P : \mathbb{R}^3 \to \mathbb{R}^{144}$ using ridge regression in a chosen feature space of the input. Let $\phi_u(u) \in \mathbb{R}^{d_u}$ denote a feature lift of the tendon command. Stacking the N_2 samples into $\Phi_U \in \mathbb{R}^{N_2 \times d_u}$ and $X_{eq} \in \mathbb{R}^{N_2 \times 144}$, we compute

$$\begin{aligned} W_P &= \arg\min_{W_P} \|\Phi_U W_P - X_{eq}\|_F^2 + \lambda_P \|W_P\|_F^2, \\ P(u) &= \phi_u(u)^\top W_P, \end{aligned} \tag{1}$$

where $\lambda_P > 0$ is the regularization weight. In practice, all states and inputs are z-score normalized on valid D_1 before regression, and we omit an explicit constant bias feature since centering makes an intercept unnecessary. The learned $P(u)$ provides a steady-state reference that can be evaluated at time-varying commands u_k during both training and rollout, anchoring predictions toward physically plausible equilibria.

Given the equilibrium prior, we define a contractive anchoring step that injects steady-state structure into the rollout. For a current predicted state $\hat{x}_k$ and the applied input u_k, we compute an anchored intermediate prediction

$$\bar{x}_{k+1} \;=\; \hat{x}_k + \beta\big(P(u_k) - \hat{x}_k\big), \qquad \beta \in (0, 1], \tag{2}$$

which pulls the prediction toward the actuation-conditioned equilibrium reference $P(u_k)$. In training we use x_k whereas in rollout we use $\hat{x}_k$. The scalar β controls the anchoring strength: smaller values yield a weaker pull that preserves more of the current predicted trajectory, while larger values enforce stronger attraction toward the equilibrium manifold. This anchoring step is designed to suppress steady-state drift during iterative rollouts, but by construction it is driven by a static map and therefore cannot reproduce transient dynamics or systematic one-step mismatch.

B. Residual Learning

The next step is to learn the residual correction from the transition dataset D_1 in a prior-consistent manner by training on anchored states $\bar{x}_{k+1}$, so the training inputs match those encountered during open-loop rollout. The residual is a feature-lifted, linear-in-parameters map fit by regularized least squares. For each transition $(x_k, u_k, x_{k+1}) \in D_1$, we compute the anchored intermediate state via (2) and define the residual target as

$$r_k^\star \;=\; x_{k+1} - \bar{x}_{k+1}. \tag{3}$$

The residual uses a regressor lift $\phi_z : \mathbb{R}^{147} \to \mathbb{R}^{d_z}$ and is formed from the anchored state and the current input

$$z_k \;=\; [\bar{x}_{k+1}; u_k] \in \mathbb{R}^{147}, \qquad R(z_k) = \phi_z(z_k)^\top W_R, \tag{4}$$

where $\phi_z(\cdot)$ is a feature lift for a single regressor z_k. Stacking all training samples gives $\Phi_Z \in \mathbb{R}^{N_1 \times d_z}$ and $R^\star \in \mathbb{R}^{N_1 \times 144}$, and we estimate W_R by ridge regression

$$W_R \;=\; \arg\min_{W_R} \; \|\Phi_Z W_R - R^\star\|_F^2 + \lambda_R \|W_R\|_F^2. \tag{5}$$

Algorithm 1 Training and rollout for the equilibrium-anchored HYBRID predictor

Input: Equilibrium dataset $D_2 = \{(u^j, x_{eq}^j)\}_{j=1}^{N_2}$; trajectory dataset $D_1 = \{(x_k, u_k, x_{k+1})\}$; ridge weights $\lambda_P > 0, \lambda_R > 0$; feature map $\phi_u(\cdot)$; feature lift $\phi_z(\cdot)$; contraction $\beta \in (0, 1]$; z-score scalers $(\mu_u, \sigma_u), (\mu_x, \sigma_x)$; optional clipping bounds.

Output: Prior parameters W_P defining $P(u_s) = \phi_u(u_s)^\top W_P$ in normalized space; residual weights W_R and rollout recursion.

(A) Training the equilibrium prior $P(u_k) \approx x_k^{eq}$
Normalize each sample: $u_s^j \leftarrow (u^j - \mu_u) \oslash \sigma_u, \quad x_{eq,s}^j \leftarrow (x_{eq}^j - \mu_x) \oslash \sigma_x$
Build regression matrices: $\Phi_U \leftarrow \big[\phi_u(u_s^1)^\top; \ldots; \phi_u(u_s^{N_2})^\top\big] \in \mathbb{R}^{N_2 \times d_u} \quad X_{eq} \leftarrow \big[(x_{eq,s}^1)^\top; \ldots; (x_{eq,s}^{N_2})^\top\big] \in \mathbb{R}^{N_2 \times n_x}$
Ridge: $W_P \leftarrow (\Phi_U^\top \Phi_U + \lambda_P I)^{-1} \Phi_U^\top X_{eq}$
Define prior: $P(u) \triangleq \phi_u(u_s)^\top W_P$ with $u_s = (u - \mu_u) \oslash \sigma_u$
(B) HYBRID residual training
Normalize all valid D_1 samples to obtain $(x_{k,s}, u_{k,s}, x_{k+1,s})$
foreach *transition* $(x_{k,s}, u_{k,s}, x_{k+1,s})$ **do**
  Compute anchored state $\bar{x}_{k+1,s}$ using Eq. (2) (with $x_{k,s}$). Form residual target $r_{k,s}^\star$ using Eq. (3). Form regressor $z_k \leftarrow [\bar{x}_{k+1,s}; u_{k,s}]$. Accumulate feature row $\phi_z(z_k)^\top$ and target row $(r_{k,s}^\star)^\top$.
Solve ridge regression for W_R using Eq. (5). Define residual model $R(z) = \phi_z(z)^\top W_R$
(C) Rollout recursion
Initialize $\hat{x}_{0,s}$ by normalizing x_0
for $k = 0$ **to** $T - 1$ **do**
  Normalize u_k to obtain $u_{k,s}$ Compute $\bar{x}_{k+1,s}$ via Eq. (2) (with $\hat{x}_{k,s}$). Update $\hat{x}_{k+1,s}$ via Eq. (6). Unnormalize $\hat{x}_{k,s}$ if needed to obtain $\hat{x}_k$
return W_P, W_R

At deployment time, the learned model generates open-loop rollouts by iterating the anchored update plus residual correction (Fig. 2(b)). Given an initial state $\hat{x}_0 = x_0$ and an input sequence $\{u_k\}_{k=0}^{T-1}$, we first compute the anchored intermediate prediction $\bar{x}_{k+1}$ from (2), then apply the residual correction to obtain the final one-step update

$$\hat{x}_{k+1} \;=\; \bar{x}_{k+1} + R([\bar{x}_{k+1}; u_k]). \tag{6}$$

This recursion is repeated for $k = 0, \ldots, T-1$ to produce a full rollout. In implementation, inputs and states are z-score normalized using statistics fit on the valid D_1 training trajectories, and we optionally saturate normalized states and inputs to fixed threshold bounds, $\tilde{x} \leftarrow \mathrm{clip}(\tilde{x}, -c, c)$, to suppress rare outliers during iterative prediction. These steps do not change the mathematical structure of (2)–(6) but improve numerical robustness when the model is rolled out over many steps under shifted actuation. Algorithm 1 summarizes the full training and rollout procedure.

C. Stability of the anchored residual rollout

A sufficient condition for boundedness and incremental stability of the anchored-residual rollout is derived next. Let the hybrid predictor update be

$$\begin{aligned} \bar{x}_{k+1} &= (1 - \beta)\hat{x}_k + \beta\, P(u_k), \qquad \beta \in (0, 1], \\ \hat{x}_{k+1} &= \bar{x}_{k+1} + R([\bar{x}_{k+1}; u_k]), \end{aligned} \tag{7}$$

where $P : \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is the equilibrium prior learned from D_2 and $R : \mathbb{R}^{n_x + n_u} \to \mathbb{R}^{n_x}$ is the residual map learned from D_1. All norms are Euclidean.

Proposition 1 (Boundedness and contraction of the rollout map)**.** *Assume the inputs satisfy $u_k \in \mathcal{U}$ for all k, where $\mathcal{U}$ is compact, and that the prior and residual are bounded on the relevant sets:*

$$\begin{aligned} \|P(u)\| &\le M_P \quad \forall u \in \mathcal{U}, \\ \big\|R([s;u])\big\| &\le M_R \quad \forall u \in \mathcal{U},\ \forall s \in \mathcal{S}, \end{aligned} \tag{8}$$

where $\mathcal{S} \subset \mathbb{R}^{n_x}$ is a set that contains the anchored states generated by (7)*. Then the predicted state sequence $\{\hat{x}_k\}$ generated by* (7) *is bounded and, in particular,*

$$\begin{aligned} \|\hat{x}_k\| &\le (1-\beta)^k \|\hat{x}_0\| + \frac{\beta M_P + M_R}{\beta}\big(1-(1-\beta)^k\big) \\ &\le \|\hat{x}_0\| + M_P + \frac{M_R}{\beta}. \end{aligned} \tag{9}$$

Moreover, if the residual is Lipschitz in its state argument uniformly over $\mathcal{U}$, i.e.,

$$\begin{aligned} \|R([s_1;u]) - R([s_2;u])\| \le L_R \|s_1 - s_2\| \\ \forall u \in \mathcal{U},\ \forall s_1, s_2 \in \mathcal{S}, \end{aligned} \tag{10}$$

where L_R is the Lipschitz constant of the learned residual map with respect to its state argument. Under this condition, the one-step map induced by (7) *is a contraction whenever*

$$(1-\beta)(1+L_R) < 1 \quad \Longleftrightarrow \quad L_R < \frac{\beta}{1-\beta}, \tag{11}$$

which implies exponential stability of the predictor recursion under fixed $u \in \mathcal{U}$.

Proof: **Boundedness.** Using (7) and applying the triangle inequality yields $\|\hat{x}_{k+1}\| \le \|(1-\beta)\hat{x}_k + \beta P(u_k)\| + \|R([\bar{x}_{k+1};u_k])\| \le (1-\beta)\|\hat{x}_k\| + \beta\|P(u_k)\| + \|R([\bar{x}_{k+1};u_k])\| \le (1-\beta)\|\hat{x}_k\| + \beta M_P + M_R$, where (8) was used. Solving this scalar recursion gives (9).

Contraction. For a fixed $u \in \mathcal{U}$, define the one-step map $F_u(\hat{x}) \triangleq (1-\beta)\hat{x} + \beta P(u) + R\big([(1-\beta)\hat{x} + \beta P(u);\, u]\big)$.

For any $\hat{x}_1, \hat{x}_2$ whose corresponding anchored states lie in $\mathcal{S}$, (10) implies $\|F_u(\hat{x}_1) - F_u(\hat{x}_2)\| \le (1-\beta)\|\hat{x}_1 - \hat{x}_2\| + L_R\|(1-\beta)(\hat{x}_1 - \hat{x}_2)\| = (1-\beta)(1+L_R)\|\hat{x}_1 - \hat{x}_2\|$.

If (11) holds, then F_u is a contraction, and the recursion is exponentially stable. ∎

Remark. In implementation, inputs are bounded by saturating tendon commands to a fixed range, and performing rollouts in normalized space. See Sec. III for implementation details.

D. Models and Baselines

Five predictors are evaluated under the same autoregressive rollout interface to isolate differences in one-step updates and stability under iteration. The proposed HYBRID method has already been explained in Sec. II-A and Sec. II-B. For comparison, we include four baselines: EQUILIBRIUM PRIOR with anchor only and no residual, RESIDUAL with residual-only lifted linear dynamics and no anchor, NEURAL PRIOR with MLP equilibrium anchor trained on D_2, and NEURAL HYBRID with neural anchor combined with the same lifted residual. These methods differ only in the one-step update used to compute $\hat{x}_{k+1}$ from $(\hat{x}_k, u_k)$. Let $\bar{x}_{k+1}$ be defined as in (2). Then:

$$\text{EQUILIBRIUM PRIOR:} \quad \hat{x}_{k+1} = \bar{x}_{k+1}, \tag{12}$$

$$\text{RESIDUAL:} \quad \hat{x}_{k+1} = \hat{x}_k + R([\hat{x}_k; u_k]). \tag{13}$$

The HYBRID update is given by (6). For neural-anchor variants, we replace $P(\cdot)$ with an MLP $f_\theta(\cdot)$ trained on the same equilibrium dataset D_2 via supervised regression while the residual learning and rollout recursion remain unchanged:

$$\text{NEURAL PRIOR:} \quad \hat{x}_{k+1} = (1-\beta)\hat{x}_k + \beta f_\theta(u_k), \tag{14}$$

$$\begin{aligned} \text{NEURAL HYBRID:} \quad \hat{x}_{k+1} = (1-\beta)\hat{x}_k + \beta f_\theta(u_k) \\ + R([\bar{x}_{k+1}; u_k]). \end{aligned} \tag{15}$$

This comparison isolates anchor parameterization: HYBRID and NEURAL HYBRID share the same residual learner and rollout recursion, differing only in whether the equilibrium map is ridge-linear or an MLP. We keep the residual linear-in-parameters to preserve the control-friendly, regularized structure used in the stability analysis; replacing it with an MLP would confound the anchor comparison. In practice, the ridge equilibrium anchor is most reliable when D_2 is small, has limited coverage, or deployment may fall outside the sampled D_2 range, as it provides a conservative and predictable anchor. The neural anchor is most useful when D_2 is large and well covered and the equilibrium map is strongly nonlinear. In this case, careful regularization and validation are needed, and performance is best when deployment inputs remain close to the D_2 distribution.

III. IMPLEMENTATION AND EVALUATION PROTOCOL

We consider a tendon-driven 3D continuum arm actuated by three tendon displacement commands $u_k \in \mathbb{R}^3$ for evaluation. The tendon inputs induce configuration-dependent strains and curvatures, producing rich three-dimensional shape evolution. All data are generated using a Cosserat-rod simulator following [20] and the discretization conventions in [21]. Fig. 3 provides an overview of the model and the backbone sampling locations used to form the discrete-time state x_k for learning.

To provide equilibrium information, we generate an equilibrium dataset $D_2 = \{(u^{(j)}, x_{\text{eq}}^{(j)})\}_{j=1}^{4520}$ by sampling constant tendon displacements $u \in \mathbb{R}^3$ and solving a static equilibrium boundary-value problem for each sample. After convergence, we extract the equilibrium state $x_{\text{eq}}^{(j)} \in \mathbb{R}^{144}$ using the same state definition as in D_1 (six backbone locations, 24 variables per location); samples that fail to converge or produce non-finite values are discarded. The remaining pairs define an empirical map $u \mapsto x_{\text{eq}}$ used to fit the equilibrium anchor.

The transition dataset $D_1 = \{(x_k, u_k, x_{k+1})\}$ consists of rollouts of length $T = 200$ at $\Delta t = 0.03$ s. For trajectory i, we store $u_{0:T-1}^{(i)} \in \mathbb{R}^{3\times 200}$ and $x_{0:T-1}^{(i)} \in \mathbb{R}^{144\times 200}$. We use 30 training trajectories ($D_{1,\text{train}}$) and 20 held-out trajectories for OOD evaluation (D_{test}). Trajectories containing non-finite inputs or states are excluded, yielding $29/30$ training trajectories and $19/20$ test trajectories used in all reported results.

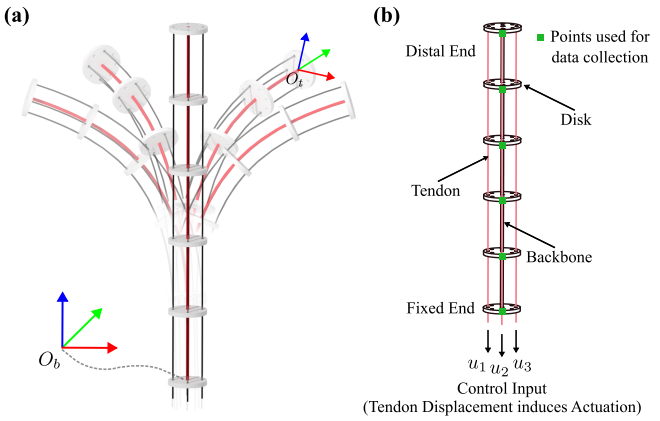

Fig. 3. **Tendon-driven continuum arm model, notation, and reference frames. (a)** Rendered view of the tendon-driven continuum arm used in this work, showing the backbone, three routing tendons, and spacer disks, with the base frame O_b and a tip frame O_t indicated. **(b)** Schematic highlighting the key components: fixed and distal ends, backbone, tendons, and disks. The control input u_1, u_2, u_3 denotes tendon displacement commands that induce actuation, and the green markers indicate the points used for data collection.

A. Input Design and OOD Evaluation

The tendon input signals are ramped, smoothed, and bounded periodic waveforms with randomized parameters. For tendon $i \in \{1,2,3\}$, we define a continuous-time command $u_i^{\mathrm{cmd}}(t) = r(t)\, a_i\, p_{m_i}\big(f_i t + \phi_i\big)$, where $r(t) = \min\Big(1, \frac{t}{t_{\mathrm{ramp}}}\Big)$, $t_{\mathrm{ramp}} = 0.5$ s is a ramp duration, $a_i > 0$ is the amplitude, $f_i > 0$ is the frequency (Hz), and ϕ_i is the phase. The waveform family is

$$\begin{aligned} p_1(\tau) &= \sin(2\pi\tau), \\ p_2(\tau) &= \sin(2\pi\tau) + 0.4\,\sin(4\pi\tau), \\ p_3(\tau) &= \frac{2}{\pi}\arcsin\big(\sin(2\pi\tau)\big), \end{aligned} \tag{16}$$

where τ denotes the waveform phase and p_3 is triangle-like. For each trajectory, we randomize (a_i, f_i, ϕ_i) and the waveform index $m_i \in \{1,2,3\}$ independently across tendons within fixed bounds, then sample at $\Delta t = 0.03$ s to obtain $u_k = u^{\mathrm{cmd}}(k\Delta t)$. Optionally, we smooth u^{cmd} with a first-order low-pass filter (time constant $\tau_f = 0.08$ s) to limit actuation bandwidth, and then saturate the commands to $u \in [-0.025, 0.025]$. The OOD test set uses the same waveform family (16) but samples its parameters from a shifted distribution that produces larger amplitudes and higher dominant frequencies than in training. This tests whether predictors remain stable and accurate under systematically more aggressive actuation at deployment.

B. Training Protocol and Hyperparameters

Training of the residual learner is performed in normalized coordinates using scalers fit on valid D_1 trajectories. The rollout updates in Eqs. (2)–(6) are executed in normalized space; predictions are unnormalized only for reporting errors in physical units. Unless stated otherwise, we use identity feature lifts, $\phi_u(u) = u$ and $\phi_z(z) = z$, set $\beta = 0.5$, fit the equilibrium anchor $P(u)$ on D_2 by ridge regression, and fit the residual on D_1 by ridge regression with regressor $z_k = [\bar{x}_{k+1}; u_k]$. Hyperparameters are selected by a small sweep over ridge penalties and feature families, and robustness is assessed over multiple random seeds by varying train/validation splits and where applicable. During rollout, we optionally apply value saturation in normalized space to inputs, states, and residual outputs to prevent numerical blow-up.

To assess sensitivity to equilibrium data, we run a D_2-fraction sweep by uniformly subsampling without replacement a fraction $\rho \in \{0.1, 0.25, 0.5, 0.75, 1.0\}$ of D_2 ($|D_2| = 4520$), fitting $P(u)$ on the subsample and training the residual on the same $D_{1,\mathrm{train}}$ transitions with an otherwise identical pipeline. Across fractions we fix $\beta = 0.5$, linear features for prior and residual, $\lambda_P = 10^{-3}$, $\lambda_R = 1$, and rollout residual clipping $\texttt{clip}_r = 2.0$. We also tested a quadratic monomial lift with no rollout improvement. Each fraction is repeated for five seeds and evaluated on the same OOD test set. We also swept $\beta \in \{0.25, 0.5, 0.75\}$ over 10 seeds; OOD RMSE was similar for $\beta = 0.5$ and 0.75 (1.901 vs. 1.890), while weaker anchoring degraded accuracy (1.950 at $\beta = 0.25$).

C. Evaluation Metrics

Multi-step rollouts are evaluated on held-out OOD trajectories using position-based shape errors from backbone points and a normalized full-state error. Let $i \in \{1, \ldots, N_{\mathrm{test}}\}$ index test trajectories and let k denote the discrete time-step index, $k = 0, \ldots, T-1$. From each state $x_k^{(i)} \in \mathbb{R}^{144}$ we extract the backbone positions at $N = 6$ arc-length samples, $p_k^{(i)} = [p_{k,1}^{(i)}, \ldots, p_{k,N}^{(i)}] \in \mathbb{R}^{3N}$ with $p_{k,j}^{(i)} \in \mathbb{R}^3$ (Fig. 3), and similarly $\hat{p}_k^{(i)}$ from the predicted rollout $\hat{x}_k^{(i)}$. All position errors are reported in physical units via a fixed length scaling factor s_{len}, for example, $s_{\mathrm{len}} = 100$ for meters-to-centimeters conversion.

We report a single scalar backbone error aggregated over the OOD set and horizon: $\mathrm{RMSE}_{\mathrm{bb}} = s_{\mathrm{len}} \sqrt{\frac{1}{N_{\mathrm{test}}\, T\, (3N)} \sum_{i=1}^{N_{\mathrm{test}}} \sum_{k=0}^{T-1} \big\| p_k^{(i)} - \hat{p}_k^{(i)} \big\|_2^2}$. For time-resolved plots we use the per-trajectory, per-time per-coordinate backbone error $e_k^{(i)} = s_{\mathrm{len}} \sqrt{\frac{1}{3N} \big\| p_k^{(i)} - \hat{p}_k^{(i)} \big\|_2^2} \in \mathbb{R}_{\geq 0}$, and summarize across trajectories at each time step using the median and interquartile range (IQR), $\mathrm{med}_i(e_k^{(i)})$ with the 25–75 percentile band. For per-trajectory distribution plots, we time-average each trajectory, $\bar{e}^{(i)} = \frac{1}{T}\sum_{k=0}^{T-1} e_k^{(i)}$, and report the distribution over i. In addition to position-based errors, we report a normalized full-state RMSE: $\mathrm{RMSE}_x = \sqrt{\frac{1}{N_{\mathrm{test}}\, T\, n_x} \sum_{i=1}^{N_{\mathrm{test}}} \sum_{k=0}^{T-1} \sum_{d=1}^{n_x} \left(\frac{x_{k,d}^{(i)} - \hat{x}_{k,d}^{(i)}}{\sigma_d + \epsilon} \right)^2}$, where σ_d is the standard deviation of state dimension d computed on the valid training set, and $\epsilon > 0$ is a small constant for numerical stability.

IV. Results

Results compare five predictors on 6 s iterative rollouts (T=200, Δt=0.03 s) over held-out OOD trajectories. Table I

TABLE I
OOD ROLLOUT RMSE OVER $T = 200$ STEPS ON $N_{\text{TEST}} = 19$ TRAJECTORIES. UNITS: CM (POSITION), DEG (ORIENTATION); STATE RMSE IS UNITLESS (NORMALIZED).

Model	State RMSE	Backbone RMSE	Tip RMSE	Tip Orientation RMSE
EQUILIBRIUM PRIOR	2.513	15.32	26.10	41.63
NEURAL PRIOR	2.515	16.08	27.51	34.44
RESIDUAL	1.943	12.21	20.62	23.06
HYBRID	1.914	11.34	19.15	21.97
NEURAL HYBRID	1.900	11.29	19.06	24.20

reports aggregate backbone position, tip position, and tip orientation RMSE. Across position metrics, adding a residual consistently improves position accuracy over anchor-only prediction: NEURAL HYBRID attains the lowest backbone and tip RMSE (11.29 cm and 19.06 cm), improving over EQUILIBRIUM PRIOR by 26.3% and 27.0%, respectively, while HYBRID achieves comparable gains (11.34 cm, 19.15 cm). Relative to RESIDUAL, the hybrid models provide a modest additional reduction in position error (about 7–8%), consistent with anchoring mainly reducing steady-state drift while the residual captures most transient mismatch. For tip orientation, HYBRID reduces RMSE from 41.63° to 24.20° (41.9%), but RESIDUAL remains best at 23.06°, suggesting the current anchor does not directly optimize orientation error under our feature and regularization choices. We find that learning the anchor does not require dense equilibrium sampling. Increasing D_2 from 10% (452 samples) to 100% (4520 samples) yields essentially no change in HYBRID OOD RMSE. The equilibrium anchor is, nonetheless, important for stable rollouts under actuation shift. We additionally evaluated identity-EDMDc/DMDc [22], N4SID [23], and linear SINDYc [24] baselines under the same OOD recursive-rollout protocol and metrics. HYBRID reduced backbone/tip RMSE by about 6.4% relative to EDMDc/N4SID and about 33% relative to SINDYc. Its residual learner also used about 28% fewer dynamic rollout transitions for training, because the equilibrium component is learned from the cheaper static dataset D_2.

Fig. 4(a) plots the median shape RMSE over time with the 25–75% band, showing error growth under iterative prediction. Fig. 4(b) summarizes the same error versus arc-length, indicating where along the backbone each model deviates most. Errors are smallest near the base and increase toward the distal end, consistent with downstream accumulation of strain and curvature deviations. Both hybrid models reduce error over time and along the backbone, with the largest gains near distal segments where drift is most pronounced. In Fig. 4(c), the end-effector coordinates (x, y, z) of a representative trajectory over a full rollout illustrate that HYBRID closely tracks the ground truth across all three axes, with remaining discrepancies appearing primarily as small amplitude and phase deviations rather than a persistent offset. Fig. 4(d) then visualizes the per-trajectory distribution of time-averaged shape RMSE. The two anchor-only predictors exhibit higher medians and wider dispersion across trajectories, consistent with sensitivity to steady-state mismatch under iterative rollout. In contrast, all

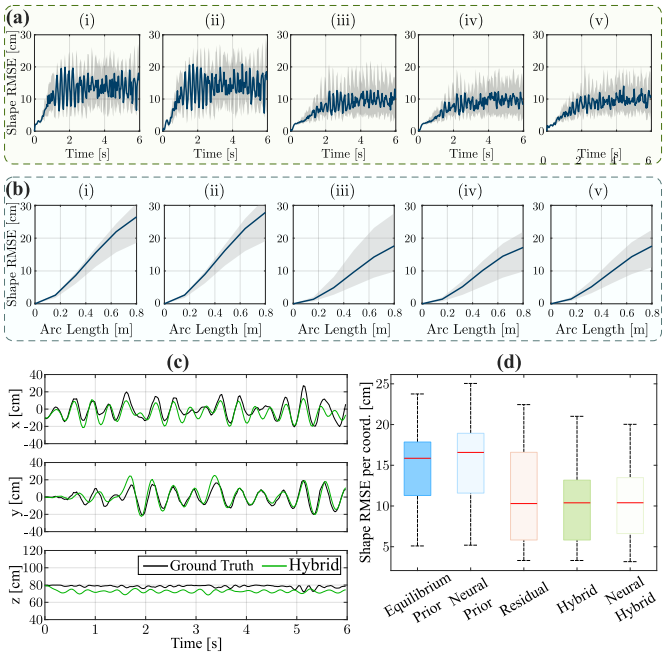

Fig. 4. **Rollout error and representative trajectory behavior on the OOD test set. (a)** Time evolution of shape RMSE over the 6 s rollout horizon for the five predictors, shown as median with 25–75% band across test trajectories. **(b)** Shape RMSE versus backbone arc length, showing how error accumulates spatially along the arm. In **(a)**–**(b)**, panels **(i)**–**(v)** correspond to EQUILIBRIUM PRIOR, NEURAL PRIOR, RESIDUAL, HYBRID, and NEURAL HYBRID, respectively. **(c)** Representative end-effector coordinate trajectories (x, y, z) comparing ground truth and HYBRID. **(d)** Boxplots of time-averaged shape RMSE over OOD test trajectories for all five predictors.

residual-based predictors reduce the median error and tighten the interquartile range, with HYBRID achieving the lowest median time-averaged shape error and a relatively compact IQR, indicating improved robustness across diverse OOD trajectories rather than gains concentrated in only a few cases. It is important to note here that the apparent difference from Table I reflects metric definition and aggregation: Table I reports aggregate normalized full-state and geometric RMSE, whereas Fig.4(d) shows the distribution of time-averaged per-trajectory shape errors.

Finally, Fig. 5 illustrates qualitative behavior under actuation shift using backbone snapshots, tip-plane trajectories, and representative failure and success cases. In Fig. 5(a)–(d), snapshots at $t = \{0, 3, 6\}$ s for four test trajectories show that EQUILIBRIUM PRIOR remains stable but exhibits steady-state mismatch, while RESIDUAL tracks transients yet can drift to an incorrect distal configuration over time. In contrast, HYBRID stays closer to the ground-truth backbone, especially near the distal end, and avoids the late-time deviations seen in the baselines. Fig. 5(e) shows top-view tip trajectories over $t \in [0, 1.5]$ s, where HYBRID more consistently matches the ground-truth planar motion than RESIDUAL. Fig. 5(f)–(g) further highlights these behaviors on trajectory 3 at $t = 3.0$ s, RESIDUAL becomes unphysical, evidenced by a violation of the constant-length constraint (backbone length growth), under iterative rollout, whereas at $t = 3.9$ s HYBRID remains well aligned with the ground-truth backbone under the same

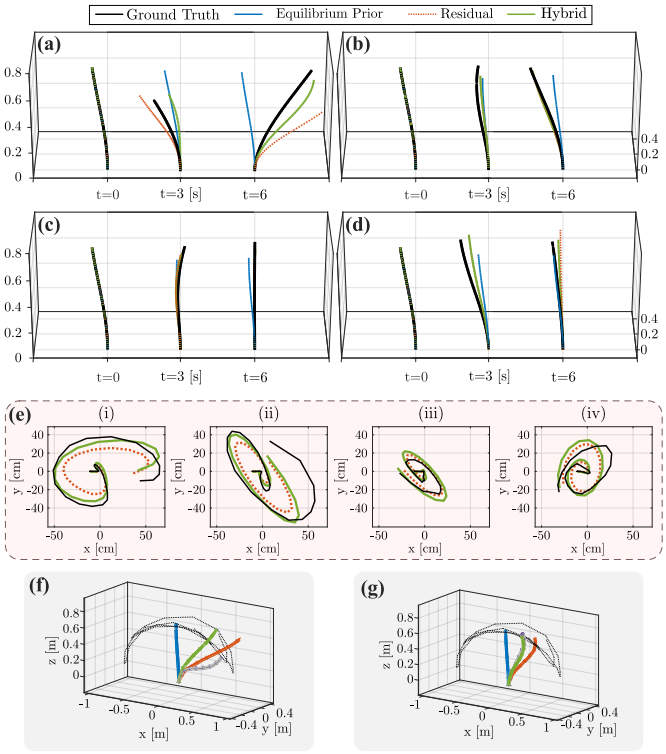

Fig. 5. **Qualitative comparison under actuation shift. (a–d)** Backbone configurations at representative times ($t = 0, 3, 6$ s) for four test trajectories: **(a)** trajectory 5, **(b)** trajectory 7, **(c)** trajectory 8, and **(d)** trajectory 15, comparing Ground Truth, EQUILIBRIUM PRIOR, RESIDUAL, and HYBRID. **(e)** Top-view tip trajectories over $t \in [0, 1.5]$ s for four trials: **(i)** trajectory 3, **(ii)** trajectory 5, **(iii)** trajectory 7, and **(iv)** trajectory 16, where HYBRID more closely follows the ground-truth planar motion than RESIDUAL. **(f)** Trajectory 3 at ($t = 3$ s) illustrating RESIDUAL rollout instability under actuation shift, where iterative prediction diverges to an unphysical configuration. **(g)** Trajectory 3 at ($t = 3.9$ s) illustrating high HYBRID accuracy under the same forcing, remaining closely aligned with Ground Truth compared to the other predictors.

forcing. These trends are consistent with Table I and Fig. 4, supporting that combining equilibrium anchoring with residual correction improves robustness under actuation shift.

A. Soft-Tail Hardware Testbed

To evaluate whether the proposed framework transfers beyond the simulated continuum-arm setting, we conducted physical experiments on a custom PneuNet soft tail driven by two antagonistic pumps (Fig. 6(a)). The hardware study follows the same two-dataset structure used throughout the paper. Static PWM–steady-angle pairs provide the equilibrium dataset for learning the input-conditioned steady response, while 0.25 Hz pump-switching trajectories provide the dynamic dataset for fitting the residual correction. Tail kinematics were extracted from video using DeepLabCut [25], as illustrated in Fig. 6(b)–(e), where the two-channel pump command $u(t) = [u_1(t), u_2(t)]^\top$ and the measured bending angle $\theta(t) = x(t)$ define the input–output pair for the hardware model.

We then tested the learned models in open loop under an actuation-frequency shift, increasing the switching frequency to 0.5 and 1 Hz while keeping the PWM level fixed. This evaluation isolates the effect of temporal distribution shift without changing the command amplitude. The aggregate error summary in Fig. 6(f) shows that HYBRID reduced mean RMSE by at least 51% relative to EQUILIBRIUM PRIOR and by at least 74% relative to RESIDUAL across the shifted-frequency tests. The rollout traces in Fig. 6(g)–(h) further reveal the underlying failure mode: RESIDUAL progressively drifts and eventually exceeds the observed bending envelope, whereas HYBRID remains bounded and tracks the measured oscillations more closely. These results show that the same equilibrium-anchoring mechanism that stabilizes continuum-arm rollouts also improves hardware prediction under frequency shift, supporting applicability beyond the original simulator-based setting.

B. Benchmark on Pendulum-Cart and Van der Pol Systems

In addition to the aforementioned implementations, we evaluate our method on two standard low-dimensional systems, pendulum–cart and Van der Pol, to stress-test recursive rollouts under controlled conditions. We compare our equilibrium-anchored HYBRID predictor against the residual-Koopman baseline of [17] using a matched protocol that preserves the baseline's prior construction and uses divergence-aware rollout metrics. Rather than claiming universally lower average error, we use these benchmarks to disentangle two practical properties: accuracy when a predictor remains stable and robustness to blow-up under identical forcing.

1) Benchmark Setup and Protocol: We consider the standard pendulum-on-a-cart system with state $x = [\phi, \dot{\phi}]^\top$, where ϕ is the pendulum angle and u is a bounded horizontal force. The dynamics are written compactly as $\dot{x} = f(x, u)$,, where $x = \begin{bmatrix}\phi & \dot{\phi}\end{bmatrix}^\intercal$, $u \in [F_{\min}, F_{\max}]$, and rollouts are evaluated against a high-accuracy RK4 simulator under identical forcing for all models. For the baseline of [17], we use their combined residual Koopman operator $\mathbf{K}_c = \mathbf{K}_p + \gamma\,\mathbf{K}_r$, where $\mathbf{K}_p$ is the Koopman prior operator (constructed as in [17]), $\mathbf{K}_r$ is a ridge-fit residual operator, and $\gamma \geq 0$ is tuned by one-step validation RMSE. Concretely, we implement the direct-encoding construction of $\mathbf{K}_p$ using a Hermite lift over $[\phi, \dot{\phi}, u]$, estimate the data-driven operator $\mathbf{K}_d$ in the same lifted space, and fit $\mathbf{K}_r$ so that $\mathbf{K}_d \approx \mathbf{K}_p + \mathbf{K}_r$ with ridge regularization, following [17]. We keep HYBRID structurally unchanged and select its hyperparameters using the same one-step validation RMSE criterion.

We evaluate open-loop rollouts under a bounded piecewise-linear random force with waypoints every 2 s and amplitude bounds $[F_{\min}, F_{\max}]$, for horizons of 4 s and 30 s. Since a model may diverge, we report completion rate (fraction of trajectories that remain finite to the horizon) and a paired shared-prefix RMSE computed over the common finite interval. For trajectory i, let $K^{\text{fail}}_{\text{HYBRID}}$ and $K^{\text{fail}}_{\mathbf{K}_c}$ denote the first non-finite step or the horizon end if no failure occurs. We define

$$K_{\text{common}}(i) = \min\Big(K^{\text{fail}}_{\text{HYBRID}},\, K^{\text{fail}}_{\mathbf{K}_c},\, K_{\text{horizon}}\Big), \tag{17}$$

and compute each model's RMSE over $k \in [0, K_{\text{common}}(i))$. We also report RMSE on the stable subset where both methods

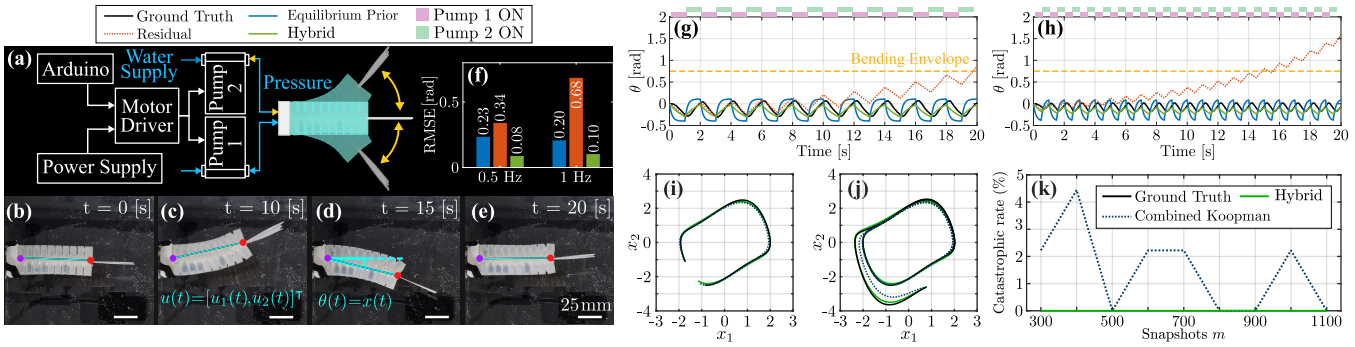

Fig. 6. **Additional validation on soft-tail hardware and the Van der Pol benchmark. (a)** Experimental setup of the antagonistic two-pump PneuNet soft tail. **(b–e)** Representative video frames used for DeepLabCut-based angle extraction, illustrating the two-pump input $u(t) = [u_1(t), u_2(t)]^\top$ and the measured tail angle $\theta(t) = x(t)$. **(f)** Mean RMSE over two runs per frequency under actuation-frequency shift. **(g–h)** Open-loop angle rollouts trained at 0.25 Hz and tested at 0.5 Hz and 1 Hz, respectively, comparing Ground Truth, EQUILIBRIUM PRIOR, RESIDUAL, and HYBRID; shaded bars indicate pump activation and the dashed line marks the bending envelope. **(i)** Van der Pol benchmark in-distribution phase portrait in the (x_1, x_2) plane. **(j)** Out-of-distribution phase portrait under harder initial conditions. **(k)** Catastrophic rollout rate versus training snapshots m, comparing the combined Koopman baseline and HYBRID.

TABLE II
PENDULUM–CART BENCHMARK UNDER MODERATE FORCING
($F_{\min} = -1$, $F_{\max} = 1$).

Metric	Hybrid	Resid. Koopman K_c
Horizon 4 s (n=10)		
Completion rate	100%	70%
Catastrophic rate	0%	30%
Shared-prefix RMSE (med.)	3.835	4.144
Win rate (shared-prefix)	60%	40%
Stable-subset RMSE (mean; n=7)	5.095	3.398
Horizon 30 s (n=10)		
Completion rate	100%	60%
Catastrophic rate	0%	40%
Shared-prefix RMSE (med.)	46.171	70.455
Win rate (shared-prefix)	50%	50%
Stable-subset RMSE (mean; n=6)	62.793	59.756

Notes: Catastrophic rate counts trials with shared-prefix RMSE 100.

complete the full horizon.

We also benchmark on the Van der Pol oscillator, $\dot{x}_1 = x_2, \quad \dot{x}_2 = \mu(1 - x_1^2)x_2 - x_1$, using the data protocol of [17] and the same residual-Koopman baseline implementation and one-step tuning procedure described in the pendulum–cart benchmark. The true parameter varies across systems as $\mu \sim \mathcal{U}[0, 1]$ (one μ per system), while the approximate-model one-step prior uses a fixed $\mu_{\text{guess}} = 0.72$. Training data are generated with $\Delta t = 0.02$ s and additive measurement noise $\sigma = 0.025$. For each system, we collect 9 trajectories of length 9 s with $x_0 \sim \mathcal{U}([-2, 2]^2)$, use a $90/10$ fit/validation split, and sweep the number of training snapshots $m \in \{300, 400, \ldots, 1100\}$ by subsampling the fit split; hyperparameters are selected by one-step validation RMSE.

For HYBRID in this benchmark, the prior is a one-step RK4 map with μ_{guess}, $x_{\text{phys}}^{k+1} = f_{\text{RK4}}(x^k; \mu_{\text{guess}})$, and the residual is learned by ridge regression under the same train/validation split and one-step selection criterion.

2) Benchmark Results and Comparison: For the pendulum–cart benchmark at moderate forcing amplitude ($F_{\min} = -1$, $F_{\max} = 1$), our Hybrid model completes all test rollouts without numerical failure with 100% completion at both 4 s and 30 s, whereas the residual Koopman baseline $\mathbf{K}_c$ fails on a substantial fraction of trajectories with 70% completion at 4 s; 60% at 30 s, and 30–40% of trials exhibiting catastrophic divergence (shared-prefix RMSE > 100). Using the paired shared-prefix metric, HYBRID attains lower median error of 3.835 against 4.144 for 4 s and 46.171 against 70.455 for 30 s, and wins on 50–60% of trajectories. On the stable subset where both methods complete the full horizon, $\mathbf{K}_c$ is often slightly more accurate with mean RMSE of 3.398 against 5.095 for 4 s and 59.756 against 62.793 for 30 s. Overall, these results indicate a tradeoff. Our method sacrifices some best-case accuracy to obtain markedly improved rollout robustness and failure avoidance under identical forcing. Additional stress tests at higher forcing amplitude ($|F| \leq 10$) reveal rapid catastrophic divergence of $\mathbf{K}_c$. Results are summarized and reported in Table II.

Next, we compare the same two predictors on the Van der Pol oscillator. Fig. 6(i)–(k) visualizes both rollout quality and robustness. In the in-distribution (ID) setting (6 s rollouts with $\mathbf{x}_0 \in [-2, 2]^2$), the phase portraits in Fig. 6(i) show that the baseline combined Koopman model K_c and HYBRID closely track the ground-truth limit-cycle geometry. Under the harder out-of-distribution initial conditions (12 s rollouts with $\mathbf{x}_0 \in [-3, 3]^2$), Fig. 6(j) highlights increased drift and deviation for K_c, whereas HYBRID remains closer to the ground truth trajectory. This robustness difference is quantified in Fig. 6(k) where HYBRID maintains a zero catastrophic-rollout rate across snapshot counts m, while K_c exhibits nonzero failure rates that depend on data availability (reaching up to 4.44%). When K_c remains stable, its accuracy can be comparable, so the advantage of HYBRID in this benchmark is primarily improved robustness rather than uniform error reduction.

V. CONCLUSION

We studied a practical failure mode in open-loop continuum-arm prediction under actuation shift in which small one-step errors accumulate during recursive rollout, bias the steady

response, and gradually amplify drift until the trajectory becomes unreliable. The proposed equilibrium-anchored residual framework addresses this failure mode by separating two roles that are often entangled in learned dynamics models. The equilibrium component provides an actuation-conditioned steady reference that counteracts gradual bias accumulation, while the residual component focuses on the remaining transient mismatch using a lightweight linear-in-parameters correction trained on the anchored rollout distribution. This prior-consistent training choice is central, because it aligns the residual's training inputs with the states encountered during iterative deployment. On the tendon-driven 3D continuum arm, the HYBRID predictor improves position accuracy while preserving numerical robustness under more aggressive actuation than used in fitting. In particular, the HYBRID predictor reduces backbone position RMSE by 26% and tip position RMSE by 27% relative to the equilibrium-only anchor, and remains stable across all tested trajectories. The soft-tail hardware testbed further supports applicability beyond the simulated continuum arm, with HYBRID reducing RMSE by at least 51% relative to EQUILIBRIUM PRIOR and 74% relative to RESIDUAL. Beyond these implementations, matched benchmarking on pendulum–cart and Van der Pol shows that the same design improves robustness relative to the combined Koopman baseline under identical forcing, especially in regimes where competing predictors can catastrophically diverge. Taken together, these results suggest a practical guideline for rollout modeling under shift. Maintaining an explicit steady reference during recursion can be as important as improving one-step accuracy, and a compact residual can then recover the missing dynamics without destabilizing the iteration.

Future work includes closed-loop evaluation in simulation and hardware, richer yet linear-in-parameters residual features, improved data coverage and backbone resolution, and online residual adaptation while retaining the equilibrium anchor for stability.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 12272008, Grant U22A2062, Grant W2442039, and Grant U23B2037 and in part by the Beijing Natural Science Foundation under Grant 3242003.

REFERENCES

[1] David A Haggerty, Michael J Banks, Ervin Kamenar, Alan B Cao, Patrick C Curtis, Igor Mezić, and Elliot W Hawkes. Control of soft robots with inertial dynamics. *Science robotics*, 8(81), 2023.

[2] Zhexin Xie, Feiyang Yuan, Jiaqi Liu, Lufeng Tian, Bohan Chen, Zhongqiang Fu, Sizhe Mao, Tongtong Jin, Yun Wang, Xia He, et al. Octopus-inspired sensorized soft arm for environmental interaction. *Science Robotics*, 8 (84), 2023.

[3] Jose Leopoldo Contreras, Ola Shorinwa, and Mac Schwager. Safe, out-of-distribution-adaptive mpc with conformalized neural network ensembles. *arXiv preprint arXiv:2406.02436*, 2024.

[4] Michael C. Yip and David B. Camarillo. Model-less feedback control of continuum manipulators in constrained environments. *IEEE Transactions on Robotics*, 30(4):880–889, 2014.

[5] Guanran Pei, Francesco Stella, Omar Meebed, Zhenshan Bing, Cosimo Della Santina, and Josie Hughes. Imu based pose reconstruction and closed-loop control for soft robotic arms. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1847–1852, 2024.

[6] Kai Huang, Ruiyong Yuan, Zenan Song, Zihao Yuan, Sheng Mao, Guoying Gu, and Feifei Chen. Differentiable ancf-based kinematic modeling of soft pneumatic robots for design optimization. *IEEE/ASME Transactions on Mechatronics*, 30(4):2826–2834, 2025.

[7] Zhiwei Wu, Jiahao Luo, Siyi Wei, and Jinhui Zhang. Unified modeling and structural optimization of multi-magnet embedded soft continuum robots for enhanced kinematic performances. *arXiv preprint arXiv:2507.10950*, 2025.

[8] Xinran Wang and Nicolas Rojas. A data-efficient model-based learning framework for the closed-loop control of continuum robots. In *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*, pages 247–254, 2022.

[9] Mohammadreza Kasaei, Keyhan Kouhkiloui Babarahmati, Zhibin Li, and Mohsen Khadem. Data-efficient non-parametric modelling and control of an extensible soft manipulator. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2641–2647, 2023.

[10] Mohammadreza Kasaei, Keyhan Kouhkiloui Babarahmati, Zhibin Li, and Mohsen Khadem. A data-efficient neural ode framework for optimal control of soft manipulators. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2700–2713. PMLR, 06–09 Nov 2023.

[11] Tom Z Jiahao, Ryan Adolf, Cynthia Sung, and M Ani Hsieh. Knowledge-based neural ordinary differential equations for cosserat rod-based soft robots. *arXiv preprint arXiv:2408.07776*, 2024.

[12] Curtis C Johnson, Tyler Quackenbush, Taylor Sorensen, David Wingate, and Marc D Killpack. Using first principles for deep learning and model-based control of soft robots. *Frontiers in Robotics and AI*, 8:654398, 2021.

[13] Tim-Lukas Habich, Aran Mohammad, Simon F. G. Ehlers, Martin Bensch, Thomas Seel, and Moritz Schappler. Generalizable and fast surrogates: Model predictive control of articulated soft robots using physics-informed neural networks. *IEEE Transactions on Robotics*, pages 1–18, 2025.

[14] Ning Tan, Peng Yu, Zhaohui Zhong, and Yunong Zhang. Data-driven control for continuum robots based on discrete zeroing neural networks. *IEEE Transactions on Industrial Informatics*, 19(5):7088–7098, 2023.

[15] Junpeng Gao, Mike Y. Michelis, Andrew Spielberg, and Robert K. Katzschmann. Sim-to-real of soft robots with learned residual physics. *IEEE Robotics and Automation Letters*, 9(10):8523–8530, 2024.

[16] Gaoming Lou, Chuang Wang, Zefeng Xu, Jiaqiao Liang, and Yitong Zhou. Controlling soft robotic arms using hybrid modelling and reinforcement learning. *IEEE Robotics and Automation Letters*, 9(8):7070–7077, 2024.

[17] Daniel Bruder, David Bombara, and Robert J Wood. A koopman-based residual modeling approach for the control of a soft robot arm. *The International journal of robotics research*, 44(3):388–406, 2025.

[18] John Irvin Alora, Mattia Cenedese, George Haller, and Marco Pavone. Discovering dominant dynamics for nonlinear continuum robot control. *npj Robotics*, 3(1):5, 2025.

[19] Zhiqiang Tang, Liying Tian, Wenci Xin, Qianqian Wang, Daniela Rus, and Cecilia Laschi. A general soft robotic controller inspired by neuronal structural and plastic synapses that adapts to diverse arms, tasks, and perturbations. *Science Advances*, 12(2):eaea3712, 2026.

[20] John Till, Vincent Aloi, and Caleb Rucker. Real-time dynamics of soft and continuum robots based on cosserat rod models. *The International Journal of Robotics Research*, 38(6):723–746, 2019.

[21] Eron Ristich, Lei Zhang, Yi Ren, and Jiefeng Sun. Physics-informed split koopman operators for data-efficient soft robotic simulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9273–9279, 2025.

[22] Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.

[23] Peter Van Overschee and Bart De Moor. N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994.

[24] Eurika Kaiser, J Nathan Kutz, and Steven L Brunton. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2219), 2018.

[25] Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie W. Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 2018.