Learning soft robotic dynamics
with active exploration

Hehui Zheng,[1,3] Bhavya Sukhija,[2] Chenhao Li,[2,3]

Klemens Iten,[2] Andreas Krause,[2,3] Robert K. Katzschmann[1,3*]

[1]Soft Robotics Lab, D-MAVT, ETH Zurich, Zurich, Switzerland
[2]Learning & Adaptive Systems Group, D-INFK, ETH Zurich, Zurich, Switzerland
[3]ETH AI Center, ETH Zurich, Zurich, Switzerland
*Corresponding author. Email: rkk@ethz.ch

Abstract

Soft robots offer unmatched adaptability and safety in unstructured environments, yet their compliant, high-dimensional, and nonlinear dynamics make modeling for control notoriously difficult. Existing data-driven approaches often fail to generalize, constrained by narrowly focused task demonstrations or inefficient random exploration. We introduce SOFTAE, an uncertainty-aware active exploration framework that autonomously learns task-agnostic and generalizable dynamics models of soft robotic systems. SOFTAE employs probabilistic ensemble models to estimate epistemic uncertainty and actively guides exploration toward underrepresented regions of the state–action space, achieving efficient coverage of diverse behaviors without task-specific supervision. We evaluate SOFTAE on three simulated soft robotic platforms—a continuum arm, an articulated fish in fluid, and a musculoskeletal leg with hybrid actuation—and on a pneumatically actuated continuum soft arm in the real world. Compared with random exploration and task-specific model-based reinforcement learning, SOFTAE produces more accurate dynamics models, enables superior zero-shot control on unseen tasks, and maintains robustness under sensing noise, actuation delays, and nonlinear material effects. These results demonstrate that uncertainty-driven active exploration can yield scalable, reusable dynamics models across diverse soft robotic morphologies, representing a step toward more autonomous, adaptable, and data-efficient control in compliant robots.

Summary: Active uncertainty-driven exploration lets soft robots autonomously learn accurate, reusable dynamics for diverse tasks.

Introduction

Problem addressed

Soft robots, with their highly deformable and compliant structures, offer compelling advantages in adaptability, safe human interaction, and environmental robustness [1]. However, these same characteristics make them extremely difficult to model and control [2]. Unlike rigid-body systems, which can be described using low-dimensional ordinary differential equations, soft robots exhibit complex, high-dimensional dynamics due to continuous deformation, distributed compliance, and nonlinear material properties [3–8]. These systems lack well-defined coordinates or rigid links, rendering traditional kinematic and dynamic formulations inapplicable or imprecise [9–11]. Instead, their behavior is governed by nonlinear partial differential equations and rich interaction effects such as viscoelasticity, hysteresis, and nonlocal actuation responses: phenomena that are not easily captured in closed-form models [6, 12].

Although data-driven models have shown promise in approximating complex dynamics, their effectiveness and generality are strongly limited by the quality and diversity of training data [13–15]. In most previous work, data are collected through task-specific demonstrations or random exploration. Task-specific data collection often results in narrow, overfitted models that generalize poorly beyond the training distribution. RANDOM exploration, on the other hand, fails to efficiently cover the vast and sparsely reachable state–action spaces characteristic of soft systems, especially in scenarios involving complex dynamics such as hybrid actuation, delayed feedback, or contact interactions. This gap between model generalization and data acquisition is particularly problematic for applications that require adaptability between tasks [16–18]. In such settings, retraining models for each new objective is impractical due to the cost and time associated with the collection of real-world data [19]. Moreover, simulation-based modeling is limited by material property discrepancies, unmodeled interactions, and slow computation, making it difficult to rely on synthetic data alone [14, 20]. Together, these issues highlight the importance of developing autonomous data acquisition strategies that can systematically and efficiently explore the dynamics of soft robots.

To address the combined challenge of poor generalization and inefficient data acquisition in soft robotic systems, we focus on autonomously learning task-agnostic, general-purpose dynamics models—models that are not only accurate, but broadly applicable to a wide range of downstream control objectives. This learning approach requires a data collection strategy that goes beyond passive or random sampling and instead actively seeks the most informative interactions with the system. The core challenge, then, is how to efficiently and autonomously explore the soft robot's state–action space in a way that exposes the full range of its dynamic capabilities, enabling robust zero-shot control across tasks and conditions without retraining. Accurately addressing this challenge is fundamental to achieving reliable autonomy and effective control in deformable robotic systems.

Objective

The objective of this work is to learn task-agnostic dynamics models for soft robots that generalize across morphologies, actuation regimes, and environmental conditions, through an exploration strategy that maximizes model coverage and informativeness. We propose an active exploration framework that uses uncertainty estimates to autonomously guide data collection toward underexplored regions of the state–action space. Our goal is to build a model that captures the full behavioral range of the robot and can be used for zero-shot planning and control across a variety of downstream tasks without retraining or task-specific adaptation. Achieving this objective requires a deeper understanding of existing modeling and learning approaches for soft robots and the role of exploration in data-driven dynamics learning.

Background and related work

Physics-based soft dynamics modeling

Modeling the behavior of soft robots is a long-standing challenge due to distributed compliance, nonlinear deformation, and complex environmental interactions inherent in soft materials [3–6]. Physics-based modeling approaches can be broadly categorized into simplified analytical models, continuum or FEM-based simulations, and reduced-order formulations. While these methods provide physically interpretable and high-fidelity representations, they face trade-offs between accuracy, generality, and computational tractability [9–11,21]. Simplified analytical models such as piecewise constant curvature (PCC) are widely used for tractable forward and inverse kinematics but often fail to capture the true behavior of physical soft robots. For example, Toshimitsu *et al.* [22] developed a PCC-based model for the pneumatically actuated SoPrA arm, reporting a notable tip position error due to elongation along the backbone and deviations from constant curvature [23]. Although accuracy can be improved by increasing the number of PCC segments or supporting variable-length sections, the growth of computational cost makes such models impractical for real-time control.
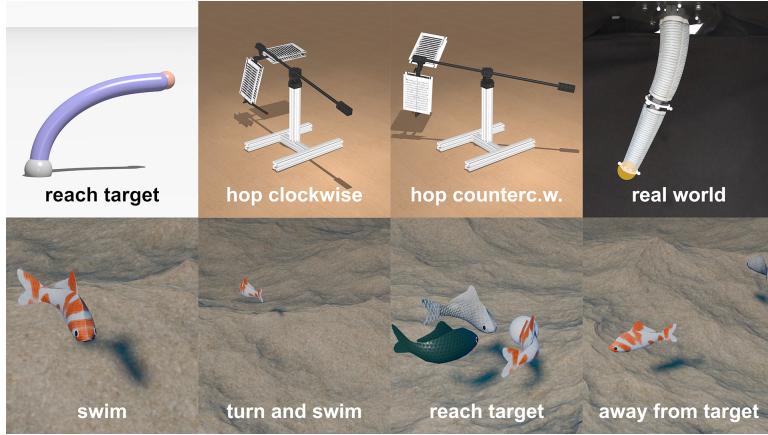
Finite element method (FEM) simulations, in contrast, can capture nonlinear material behavior and distributed deformation but remain difficult to parameterize for real-world soft robots composed of heterogeneous materials. Fiber-reinforced and tendon-driven designs introduce anisotropic stiffness and nonlinear coupling that require fine-grained meshing and complex constitutive models, significantly increasing simulation cost [24–26]. Simplified abstractions—such as modeling helical fiber nets as high-stiffness spring rings—are often used [27], but these neglect internal reinforcements critical for capturing dynamic interactions [28,29]. Consequently, FEM models can approximate quasi-static configurations but often fail under fast actuation or high loads. Reduced-order models have been proposed to balance fidelity and tractability, yet they typically sacrifice generality and robustness across different tasks and environments [30–32].

Data-driven soft dynamics modeling

Data-driven approaches that overcome these limitations have become increasingly popular. Recent work has applied supervised learning techniques—such as neural networks, Gaussian processes, and Koopman operator theory—to learn soft robot dynamics directly from sensorimotor experience [33–37]. These methods have demonstrated strong empirical performance on specific tasks such as locomotion, grasping, or shape estimation. However, the generalizability of these models is highly dependent on the quality and diversity of the training data. In particular, dynamics learned from task-specific demonstrations tend to overfit to the distribution of states and actions visited during those tasks, making them unreliable for use in novel scenarios. A complementary line of research has investigated hybrid modeling, where physical models are combined with learned components—for instance, learning residual dynamics on top of a simplified analytical model [13,14,38–41] or utilizing the physical model as a prior for Bayesian deep learning [42]. While such models can improve sample efficiency and interpretability, they still suffer from the same core limitation: the training data must sufficiently cover the relevant state–action space for the model to generalize.

Active exploration in dynamics learning

Recent work in robot learning has emphasized the importance of exploration in acquiring diverse and informative data for model learning. Techniques based on intrinsic motivation, information gain, and

Movie 1: SoftAE: Generalizable active exploration framework for efficient soft robotic dynamics learning and zero-shot control. Video available at: `https://youtu.be/kA2Rj6cDkpw`

uncertainty sampling have been employed in reinforcement learning, system identification, and dynamics model acquisition [43–47]. These approaches have shown promise in improving generalization, especially in rigid-bodied robots and simulators [16–18]. However, they remain underexplored in the context of soft robotics, where the state and action spaces are high-dimensional, often partially observable, and coupled in complex, non-intuitive ways. Some works have applied curiosity-driven exploration or ensemble-based uncertainty estimation to guide learning in high-dimensional or underactuated systems [16, 47–51]. Yet, few studies have explicitly targeted soft robotic platforms, where compliant body dynamics and sparse sensing characteristics pose unique challenges [52–54]. Moreover, the evaluation of these methods has largely been limited to simulated environments or rigid systems, leaving open questions about their robustness under real-world conditions and sensor noise. No prior work systematically studies uncertainty-driven exploration in physically compliant systems.

Contributions

This work makes three primary contributions toward data-efficient learning of generalizable dynamics models for soft robotic systems (Movie 1). First, we propose SOFTAE, an active exploration framework that combines optimism tailored to the unique challenges of soft robotics. Our approach leverages model uncertainty as an intrinsic motivation signal to autonomously guide exploration toward regions of the state–action space where the current dynamics model is least certain. This shifts data collection away from passively following task-specific trajectories or random perturbations and instead focuses on maximizing information gain during training. By actively seeking out underrepresented dynamics, our method enables a more efficient and comprehensive coverage of the robot's capabilities.

Second, we demonstrate that the resulting dynamics models are task-agnostic and highly generalizable, enabling effective zero-shot planning for previously unseen control objectives. Rather than requiring additional data or fine-tuning when faced with new tasks, the learned model, trained purely through exploratory interaction, can be directly used in downstream motion planning. This ability to decouple model learning from task specification is especially valuable in soft robotics, where manual reconfiguration and retraining are costly and time consuming. Our results show that exploration driven by model uncertainty leads to broad functional coverage of the robot's operational domain, which directly improves

4

downstream performance on tasks not seen during training.

Finally, we provide extensive empirical validation of our approach, both in simulation and in real soft robotic hardware. We evaluate performance across multiple task domains and show that our exploration strategy yields dynamics models that outperform baselines in prediction accuracy, generalization, and zero-shot control success. These experiments highlight not only the feasibility of our method but also its robustness to real-world complexities such as sensor noise and actuation uncertainty. Overall, our work provides a practical and scalable framework for building reusable soft robot models and represents a step toward more autonomous and adaptable learning systems in embodied robotics.

Results

We evaluate the proposed active exploration pipeline across both simulated and real-world soft robotic systems with highly nonlinear dynamics (Figure 1). Specifically, we assess whether the learned models (i) capture the full behavioral range of each system, (ii) enable accurate zero-shot planning across multiple downstream tasks, and (iii) can be obtained reliably by a unified pipeline that generalizes across soft robotic morphologies and control regimes. We first demonstrate results in simulation, and then validate on real hardware.

In simulation, we implement three distinct soft robotic systems as shown in Figure 1, A–C: a continuum arm modeled as a Cosserat rod, an articulated fish with deformable skin swimming in water, and a musculoskeletal leg actuated by electrohydraulic muscles and a direct current (DC) motor. Each environment presents unique challenges, from continuous elastic deformation in the arm, to two-way fluid–structure interaction in the fish, and hybrid muscle-driven actuation with ground contact in the leg, and also requires fundamentally different actuation modalities. The associated tasks also vary substantially in structure, from fixed-base reaching to aquatic locomotion to contact-rich hopping. To assess real-world applicability, we apply the same exploration strategy to the pneumatically actuated SoPrA arm [22], shown in Figure 1D. Across all environments, the learned dynamics models are evaluated on multiple downstream control tasks, without task-specific retraining.

Baseline Exploration and Learning Methods

We compare our proposed active exploration method (**SOFTAE**) against two representative baselines, designed to test the value of active exploration and task-agnostic learning (Figure 1E). As an undirected baseline, **RANDOM** samples actions uniformly from the action space, serving as a reference for the benefit of uncertainty-guided exploration. To assess the value of task-agnostic exploration for downstream generalization, we compare against **H-UCRL** [55], a model-based reinforcement learning (RL) method trained specifically on individual downstream tasks. Across all methods, we employ probabilistic ensembles (PEs) [56] to model dynamics and use model predictive control (MPC) with the improved Cross-Entropy Method (iCEM) optimizer [57] for the planning of control actions.

Zero-shot Task Performance Across Simulated Soft Robotic Systems

To systematically evaluate generalization, we consider three simulated systems of diverse physical and control complexity: a continuum arm, an articulated fish with deformable skin immersed in fluid, and a musculoskeletal leg actuated by electrohydraulic muscles and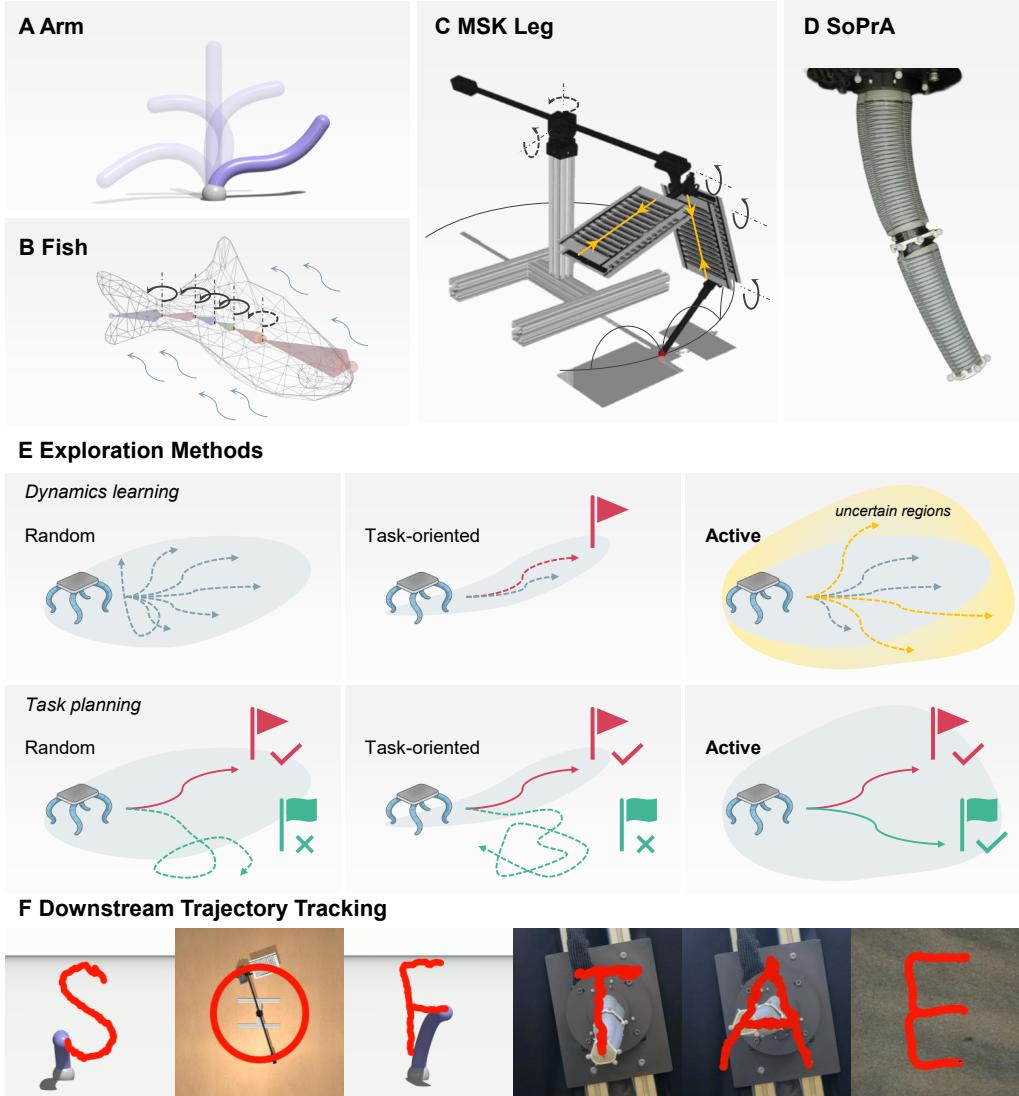 a DC motor (Figure 1, A–C; Movie S1–S3 available at: S1 – https://youtu.be/8hvBvkHiU0g, S2 – https://youtu.be/7AykVWWxQq0, S3 –

Figure 1: **Learning soft robotic dynamics with active exploration.** (**A–D**) Soft robotic platforms studied in this work: (A) soft continuum arm, (B) deformable fish in fluid, (C) musculoskeletal (MSK) leg with electrohydraulic actuation (yellow arrows), and (D) real-world pneumatically actuated arm (SoPrA). Their nonlinear, high-dimensional, and deformable dynamics pose significant challenges for generalizable model learning. (**E**) We address this with SOFTAE, an active exploration strategy that autonomously collects informative and diverse data for training dynamics models. During dynamics learning (top), RANDOM exploration is unguided, and task-oriented method overfits. SOFTAE instead targets regions of high model uncertainty, improving data coverage and model accuracy. This enables robust generalization to diverse downstream tasks (bottom), as (**F**) demonstrated across all platforms.

https://youtu.be/oY4g1fq6lM4). These environments span a wide range of dynamic behaviors, from continuous elastic deformation to two-way fluid–structure interaction and hybrid actuation with ground contact. Table 1 summarizes the task definitions and state–action space dimensions. Full

Table 1: **Simulated soft robotic environments for dynamics learning.** The table summarizes three representative environments investigated in this work: soft continuum arm, articulated fish with deformable skin in fluid, and musculoskeletal leg, together with their associated tasks. For each environment, the state space dimensions correspond to the size of the observation vector, while the action space dimensions denote the number of independent actuation signals. Detailed task rewards and decomposition of state–action space are provided in [58].

Environment	Task	State Space Dimensions	Action Space Dimensions
Soft Continuum Arm	*(i)* reach close target	58	12
	(ii) reach far target	58	12
Articulated Fish in Fluid	*(iii)* swim along +x direction	15	4
	(iv) swim along -x direction	15	4
	(v) swim to target	18	4
	(vi) swim away from target	18	4
Musculoskeletal Leg	*(vii)* hop counterclockwise	10	5
	(viii) hop clockwise	10	5

descriptions are provided in the Materials and Methods section.

We evaluated downstream control performance across all eight tasks using the learned dynamics models without any additional task-specific fine-tuning. Our active exploration method (SOFTAE) and the RANDOM baseline are tested in a zero-shot setting, while the task-specific model-based RL baseline (H-UCRL) is trained only on tasks *(i)*, *(iii)*, *(v)*, and *(vii)*. The remaining tasks are held out during training and serve as unseen generalization tests. Each method is evaluated across 10 random seeds for the tasks *(i)–(ii)* and 5 seeds for the others.

As shown in Figure 2, SOFTAE substantially outperforms RANDOM, with the gap being most pronounced in environments where effective behavior depends on coordinated actuation under nonlinear or delayed dynamics. For instance, swimming requires synchronized undulation of the spine against delayed fluid feedback from viscous drag, inertia, and vortex shedding, while hopping demands precise muscle-motor coordination and accurate timing of ground contact. Lacking such structured patterns, RANDOM exploration seldom produces meaningful motion, preventing it from discovering viable behaviors and leading to failure on tasks *(v)–(viii)*.

Across all tasks, SOFTAE matches or exceeds the performance of H-UCRL on those for which it was trained, and significantly outperforms on held-out tasks where H-UCRL fails to generalize. Notably, SOFTAE surpasses H-UCRL even on tasks *(v)* and *(vii)* (fish swimming and leg hopping), despite H-UCRL being trained directly on these objectives. These results underscore the limitations of task-specific exploration, which often overlooks coordinated control patterns not easily discovered by local optimization around a single objective. In contrast, SOFTAE's uncertainty-driven exploration strategy collects diverse data that enables robust, zero-shot control across morphologies and tasks.
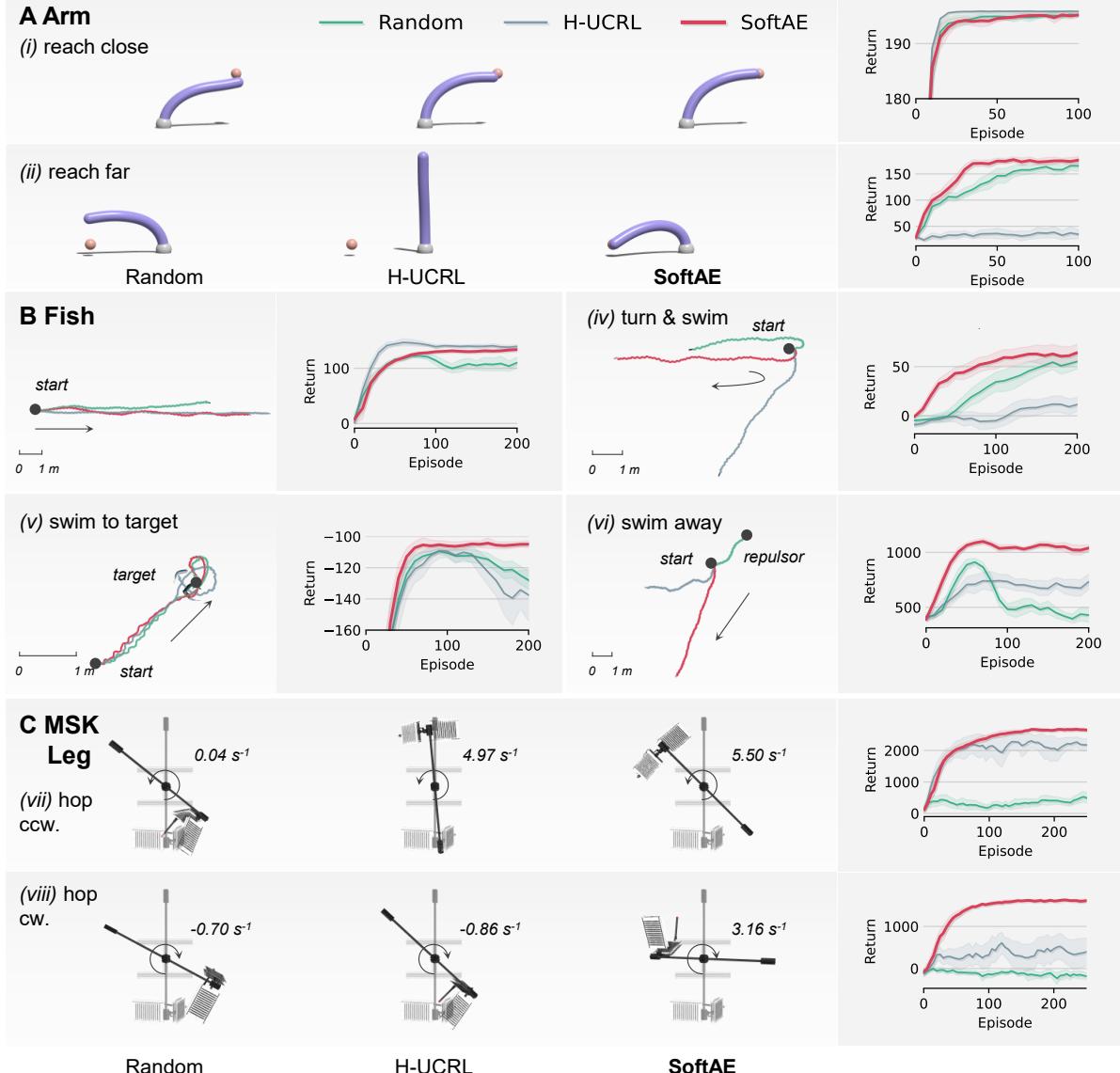
Figure 2: **Active exploration enables robust zero-shot task performance across diverse soft robotic systems.** We evaluate SOFTAE against baseline exploration strategies across three soft robotic platforms: (**A**) a soft continuum arm, (**B**) a deformable fish swimming in fluid, and (**C**) a musculoskeletal (MSK) leg actuated by electrohydraulic muscles and a DC motor. H-UCRL, a task-specific model-based baseline, is trained only on tasks *(i)*, *(iii)*, *(v)*, and *(vii)*, while the remaining tasks are held out as unseen. Across tasks, SOFTAE consistently produces task-aligned and physically plausible behaviors, while RANDOM and H-UCRL often fail to generalize to the hard or unseen scenarios. Returns are averaged over 10 random seeds for tasks *(i)* and *(ii)*, and 5 seeds for the remaining tasks.

Data Diversity Through Active Exploration

The strong downstream performance of SOFTAE suggests that its advantage comes from collecting training data that is broader in coverage and more informative than baselines. To test this hypothesis, we
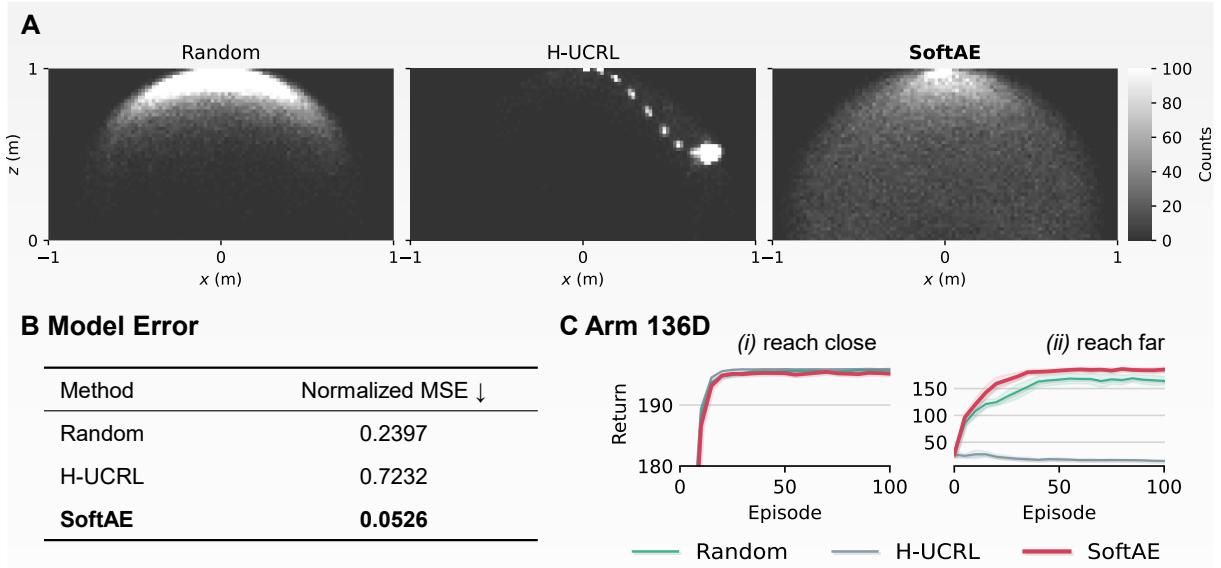
Figure 3: **Active exploration improves data coverage, model accuracy, and scales to high-dimensional dynamics. (A)** Spatial coverage of collected data for the soft continuum arm, visualized by projecting tip positions onto the x–z plane. Heatmaps show visitation frequency (counts) per spatial bin. SOFTAE explores more broadly and uniformly across the reachable workspace compared to RANDOM and H-UCRL, enabling the collection of diverse and informative training data for dynamics learning. **(B)** Normalized mean squared error (MSE) of learned dynamics models on a held-out set of 17,451 transitions. SOFTAE achieves the lowest error, indicating more globally accurate models. **(C)** Return curves for downstream task performance on a soft arm with a 136-dimensional state space, reaching *(i)* close and *(ii)* far targets. SOFTAE maintains high performance even in this high-dimensional setting, matching H-UCRL on its trained task and outperforming baselines on the more challenging unseen task. Returns are averaged over 10 seeds.

examine the exploration behavior in the continuum arm environment (Figure 1A). Because the full state space is 58-dimensional and cannot be directly visualized, we instead project the 3D tip position onto a 2D plane and aggregate the resulting coordinates into workspace heatmaps (Figure 3A).

Compared to the RANDOM baseline, which concentrates samples near the rest state, SOFTAE achieves far more uniform coverage across the reachable workspace. In contrast, task-specific H-UCRL gathers data narrowly along its training trajectory, underscoring its lack of task-agnostic exploration. These results show that active exploration systematically directs the robot toward underexplored regions, producing broader and more balanced data distributions. Such diversity provides richer supervision for dynamics learning, which we next assess through model accuracy.

Model Accuracy with Collected Data

To quantify how exploration diversity impacts dynamics model learning, we evaluate the accuracy of dynamics models trained on data from each method. Specifically, we compute the mean squared error (MSE) between predicted and ground-truth next states over a held-out validation set, which consists of rollouts toward 500 target positions uniformly sampled across the workspace. Each rollout is trimmed

9

upon target reach to avoid oversampling near-goal regions, resulting in a total of 17,451 transitions. To ensure comparability across state dimensions with varying scales, we normalize each dimension by its standard deviation before computing the MSE.

As reported in Figure 3B, SOFTAE's broader and more balanced exploration yields significantly lower next-state prediction errors across the reachable workspace. In other words, by targeting regions of high uncertainty, active exploration improves data coverage and directly translates into more accurate dynamics models, which form the foundation for the improved zero-shot task performance observed earlier.

Scalability to High-Dimensional Dynamics

For soft robotic systems that exhibit continuous and complex deformation, the resulting state representations are often high-dimensional, posing a challenge for model learning and planning. To evaluate whether our exploration strategy scales to such settings, we revisit the continuum arm environment from tasks *(i)* and *(ii)* and increase the resolution of the state representation. Instead of observing five discrete points along the arm, we extract features from eleven evenly spaced points, resulting in a 136-dimensional state space. This modification isolates the effect of higher observation dimensionality while keeping the action space and task objectives identical to the original setup.

As shown in Figure 3C, SOFTAE maintains high performance on both reaching tasks, achieving fast learning and high final reward despite the increased dimensionality. In contrast, the RANDOM baseline performance slightly degrades, highlighting the value of uncertainty-aware exploration in guiding data collection in large state spaces.

Real-World Active Exploration with a Pneumatically Actuated Arm

Having validated SOFTAE across multiple simulated systems, we next assess its performance on a physical soft robotic platform. Modeling such continuum systems in the real world is particularly challenging due to continuous deformation, nonlinear material properties, and coupled pneumatic actuation. To examine whether our data-driven exploration framework can handle these complexities, we evaluate it on the pneumatically actuated SoPrA arm [22], a two-segment continuum arm constructed from a continuous silicone shell with six internal chambers as shown in Figure 4A and Movie S4 available at `https://youtu.be/JRQ-4fM1yVc`. Each segment is assembled by combining three individually fabricated fiber-reinforced chambers, and all six chambers can be independently pressurized to generate complex, spatially distributed deformations for 3D motion. This setup provides a representative example of high-compliance soft manipulators commonly used in physical human-robot interaction, soft grasping, and wearable robotics [59, 60].

The system is actuated through differential pressure commands, resulting in a 6-dimensional action space corresponding to the pressure change in each chamber. The arm's state is captured via a motion capture system that tracks three removable plastic rings with reflective markers, mounted at the base, midpoint, and tip of the arm (Figure 4A). For the midpoint and tip rings, we observe the 3D position, linear velocity, orientation (as a quaternion), and angular velocity, all with respect to the base ring, to account for the variability in mounting. We also include the absolute pressure values commanded to all six chambers at each step, yielding a total state space of 32 dimensions.

For downstream evaluation, we define two 3D target-reaching tasks, labeled as tasks *(ix)* and *(x)*, where the objective is to move the arm's tip to a fixed target position in Cartesian space. These tasks are designed to mirror the simulated soft arm reaching tasks *(i)* and *(ii)*: in task *(ix)*, the target lies near the arm's rest
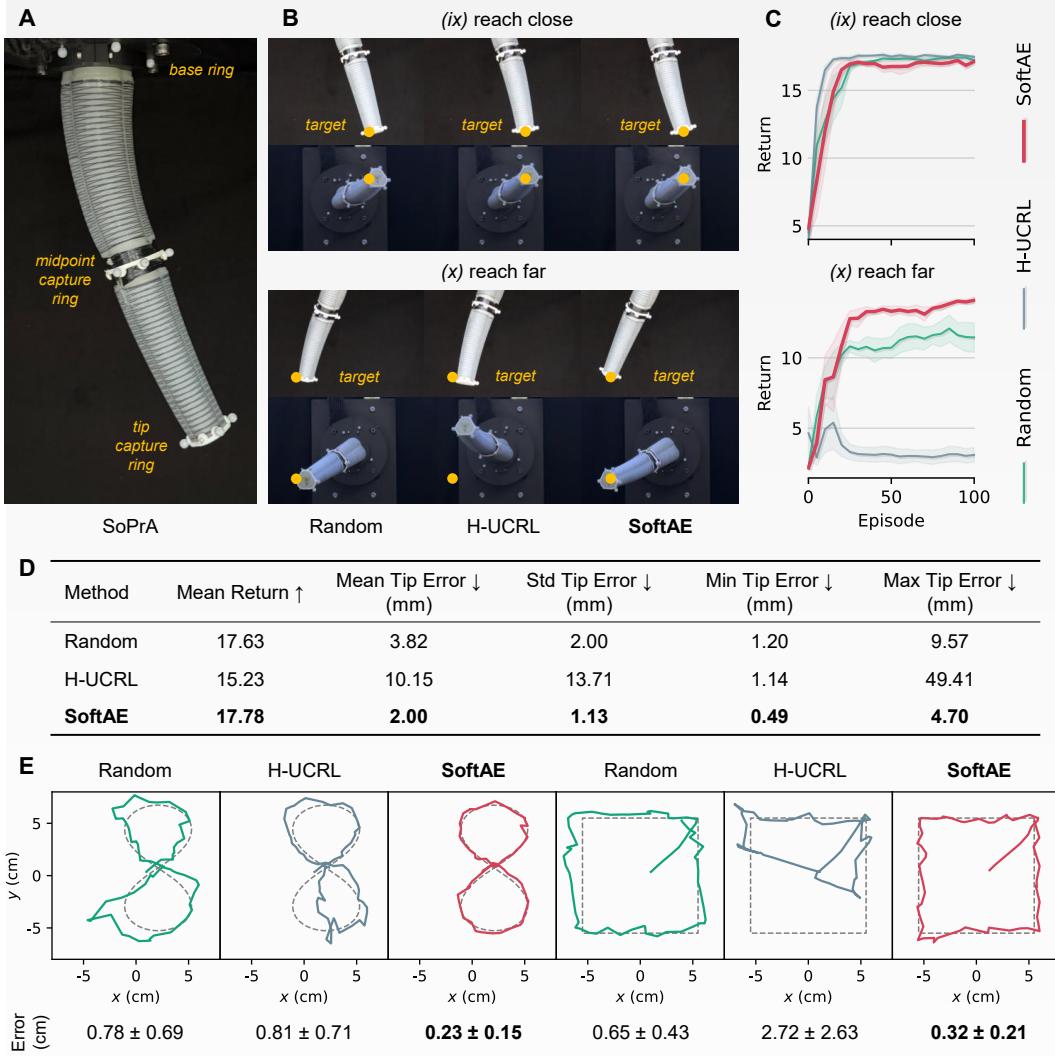
Figure 4: **Real-world dynamics learning and task performance on a pneumatically actuated soft arm.** (**A**) SoPrA: a fiber-reinforced, pneumatically actuated continuum arm with motion capture via marker rings. (**B**) Reaching tasks for *(ix)* close and *(x)* far targets. Top: front view; bottom: bottom view. Models are trained using RANDOM, H-UCRL (task-specific, trained only on *(ix)*), or SOFTAE (ours). Yellow dots indicate target positions. (**C**) Return curves show all methods perform similarly on task *(ix)*, while SOFTAE outperforms on the harder, unseen task *(x)*. Returns are averaged over 3 random seeds, shaded regions denote standard deviation. (**D**) Quantitative evaluation over the same set of 20 random targets shows that SOFTAE achieves the highest mean return and lowest tip position error compared to baselines. (**E**) Trajectory tracking performance on two reference trajectories: square (top row) and lemniscate (bottom row). SOFTAE tracks both shapes more accurately than baselines.

configuration, while in task *(x)*, the target is positioned closer to the edge of the reachable workspace, requiring greater deformation. Successful zero-shot planning in this real-world setting demands robustness to sensor noise, actuation delays, and physical nonidealities such as compliance, hysteresis, and pneumatic

variability.

Due to hardware constraints, including the inability to parallelize experiments and the time required for safe data collection, we evaluate real-world performance using 3 random seeds. After a fixed exploration budget, SOFTAE and RANDOM each learn a dynamics model, which is then used for zero-shot planning on tasks *(ix)* and *(x)*. In contrast, H-UCRL is trained directly on task *(ix)* and then evaluated on the unseen generalization task *(x)*. As shown in Figure 4, B and C, all methods achieve comparable performance on the simpler reaching task *(ix)*, where the target lies near the arm's rest configuration. In this scenario, even random exploration suffices to collect informative data for learning the local dynamics model. However, in the more challenging task *(x)*, which involves reaching toward the edge of the workspace, SOFTAE outperforms both RANDOM and H-UCRL.

To assess generalization in real-world conditions, we evaluate all methods on the same set of 20 randomly sampled 3D target positions within the arm's reachable workspace (Figure 4D). SOFTAE achieves the highest mean return and the lowest average tip error of $2\,\mathrm{mm}$ across all targets, indicating both time-efficient reaching behavior and high positional accuracy across diverse target positions. This level of error is well within practical tolerances for soft arm manipulation tasks such as reaching and positioning, where centimeter-level accuracy is usually sufficient due to the compliance and safety of the manipulator. While RANDOM exploration performs reasonably well on close targets, it suffers from higher variability due to inconsistent coverage of the workspace. H-UCRL exhibits the highest maximum tip error ($49.41\,\mathrm{mm}$), over ten times that of SOFTAE ($4.70\,\mathrm{mm}$), underscoring the brittleness of task-specific training when deployed beyond its scope. In contrast, SOFTAE's uncertainty-guided exploration yields a more robust and generalizable model that maintains accuracy throughout the workspace. Such consistency is critical for real-world deployment of soft robotic systems, where new tasks may arise without retraining and safety margins depend on reliable worst-case performance. We additionally evaluate performance on continuous trajectory tracking, using both square and lemniscate reference paths (Figure 4E). SOFTAE tracks both trajectories more accurately than the baselines, which often fail in regions where their models lack sufficient training coverage.

Discussion

Key Findings and Implications

This work addresses the fundamental challenge of learning generalizable dynamics models for soft robotic systems, which are often characterized by high-dimensional, nonlinear, and history-dependent behaviors. We propose an active exploration framework that leverages epistemic uncertainty to guide data collection toward underexplored and informative regions of the state–action space. Through systematic evaluation across simulated platforms, including a soft continuum arm, an articulated fish robot, and a musculoskeletal leg, we demonstrate that this approach enables efficient and scalable model learning across diverse soft robotic morphologies and control regimes. The resulting dynamics models support robust zero-shot control across multiple downstream tasks without requiring task-specific retraining. We further validate the method on a pneumatically actuated continuum arm in the real world, demonstrating reliable performance under sensing noise, actuation delays, and complex material behaviors.

These findings highlight the practical importance of data-efficient exploration in the field of soft robotics, where accurate simulation is often unavailable, a large amount of data is required to capture the continuous high-dimensional state–action space, and real-world data collection is expensive and time-consuming. Unlike the random exploration method, which fails to scale with the size of the state–action space, or

task-specific model-based approaches that overfit to the task region, our method autonomously focuses data collection in underexplored regions, enabling broader model coverage with fewer samples. Moreover, because soft robotic systems often exhibit continuous deformation and effectively infinite degrees of freedom, their state representations are inherently high-dimensional. In practice, these must be discretized or feature-extracted at increasing resolution, which substantially raises the cost of model learning and planning. The demonstrated scalability of SOFTAE to such settings indicates that uncertainty-driven exploration remains effective even as observation complexity increases, underscoring its potential for deployment on soft robots with rich sensory feedback or fine-grained state estimation. Together, these results establish SOFTAE as a promising foundation for scalable, autonomous learning in soft robotics, across platforms, tasks, and reliable in real-world environments.

Possible Extensions

While the proposed active exploration method shows robust performance across a variety of soft robotic systems, several limitations remain. First, although the method is scalable to moderately high-dimensional state spaces, its performance in very high-dimensional sensory modalities, such as vision or tactile images, remains to be investigated. We anticipate that extending active exploration to raw sensory inputs can be achieved by combining epistemic uncertainty with learned latent state representations, as demonstrated in [45, 61]. Second, the current implementation assumes access to relatively accurate proprioceptive and state measurements. In future work, integrating multimodal sensing (e.g., RGB-D, tactile, or proprioception with noise models) could improve robustness and applicability in less controlled settings. Additionally, SOFTAE currently operates in episodic settings with fixed-length exploration budgets and offline model learning. Prior works have theoretically and empirically shown that uncertainty-based exploration can also be extended to the non-episodic setting [61, 62]. Enabling closed-loop, online model refinement during long-horizon deployments would further improve adaptability. Another promising direction is to integrate low-fidelity physical models as priors, which has been shown to yield orders-of-magnitude improvements in sample efficiency [42].

One practical limitation of our current implementation is the use of iCEM optimizer for model predictive control. While sample-efficient, it can be computationally demanding and less responsive in high-frequency control scenarios. For tasks involving sparse rewards, long-term planning horizons, or rapid adaptation to dynamic environments, an alternative is to integrate SOFTAE with model-based reinforcement learning (MBRL) [63–65], where simulated rollouts from the learned dynamics model are used to train a policy. This hybrid approach could provide more responsive control during deployment while retaining uncertainty-aware exploration during data collection. However, MBRL can suffer from instability due to compounding model errors, making it less reliable as a general replacement for trajectory optimization. To demonstrate the consistency of our approach across different control paradigms, we evaluate a model-based policy optimization variant of SOFTAE in the Supplementary Materials [58]. The results show comparable performance and lead to the same overall conclusion: active exploration yields broader coverage and better generalization than RANDOM exploration or task-specific learning.

Materials and Methods

Active Exploration of State-Space Models

We study a general discrete-time nonlinear dynamical system of the form $s_{t+1} = f^*(s_t, a_t) + w_t$, where $s_t \in \mathcal{S} \subseteq \mathbb{R}^{d_s}$ is the state, $a_t \in \mathcal{A} \subseteq \mathbb{R}^{d_a}$ the control action, and $w_t \in \mathcal{W} \subseteq \mathbb{R}^{d_s}$ the process noise. The

system dynamics $\boldsymbol{f}^*$ are unknown and we aim to learn them from data. To this end, we consider the episodic RL setting with episodes $n \in \{1, \ldots, N\}$. At the beginning of episode n, we select and roll out a policy $\boldsymbol{\pi}_n : \mathcal{S} \mapsto \mathcal{A}$ for a horizon of T steps on the true system. We then use the data collected from the rollout $\boldsymbol{\tau}^{\boldsymbol{\pi}_n}$ to learn an estimate $\boldsymbol{\mu}_n$ of $\boldsymbol{f}^*$. However, when learning an unknown function, simply obtaining a single estimate is often insufficient since it does not quantify our lack of knowledge about the true function. In particular, in areas of the state–action space where we have limited data, we would expect our estimate to be less accurate compared to areas where we have collected data in abundance. To capture our confidence about $\boldsymbol{\mu}_n$, we additionally estimate the uncertainty $\boldsymbol{\sigma}_n$ around its predictions. Intuitively, in regions where we have less data, we would expect the uncertainty to be high, and low in regions where we have collected lots of data. There are several uncertainty estimation models that can be used for this purpose, the most classical one being Gaussian process regression [66]. However, Bayesian deep learning models, in particular ensembles, are also often used for uncertainty quantification [56]. Similar to prior work [17, 45, 67], we use probabilistic ensembles for uncertainty quantification in this work.

Thus far we have discussed how we leverage the data collected from the rollouts to learn an uncertainty-aware model of the underlying dynamics. In the following, we discuss how the policy $\boldsymbol{\pi}_n$ is selected during each episode for data collection. The goal of our algorithm is to approximate $\boldsymbol{f}^*$ over the reachable state–action space. To this end, we study an active exploration setting using an intrinsic reward function based on the epistemic uncertainty about $\boldsymbol{f}^*$ as advocated by Sukhija *et al.* [17, 61]. To efficiently explore the system dynamics, one tempting approach would be to optimize for the following objective:

$$\boldsymbol{\pi}_n^* = \arg\max_{\boldsymbol{\pi} \in \Pi} J_n(\boldsymbol{\pi}) = \arg\max_{\boldsymbol{\pi} \in \Pi} \mathbb{E}_{\boldsymbol{\tau}^{\boldsymbol{\pi}}} \left[\sum_{t=0}^{T-1} \| \boldsymbol{\sigma}_n(\boldsymbol{s}_t, \boldsymbol{\pi}(\boldsymbol{s}_t)) \| \right], \tag{1}$$

$$\boldsymbol{s}_{t+1} = \boldsymbol{\mu}_n(\boldsymbol{s}_t, \boldsymbol{\pi}(\boldsymbol{s}_t)) + \boldsymbol{w}_t. \tag{2}$$

The reward in Equation (1) encourages the agent to explore regions with high uncertainty, since high uncertainty regions typically correspond to regions where we have less data. Thus, by maximizing the uncertainty of our model, we encourage the agent to efficiently cover the state–action space. To obtain a policy, any policy or trajectory optimization technique, e.g., iCEM [57] can be used. Once we have learned the underlying dynamics well, we can leverage our model $(\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n)$ for analyzing and controlling the system. Moreover, given any reward function, we can solve the underlying task by using our learned model for planning.

Observe that in Equation (2) above, we use the mean model $\boldsymbol{\mu}_n$ for planning the state propagation. Combined with the active exploration objective, we call this algorithm MEAN-AE. Buisson-Fenet *et al.* also propose a similar approach [68]. However, Chua *et al.* found that planning with the mean often underperforms in practice since the policy exploits the inaccuracies in the mean model [67]. Instead, they propose a trajectory sampling approach for uncertainty propagation, which we adopt as follows:

$$\boldsymbol{\pi}_n^* = \arg\max_{\boldsymbol{\pi} \in \Pi} J_n(\boldsymbol{\pi}) = \arg\max_{\boldsymbol{\pi} \in \Pi} \mathbb{E}_{\boldsymbol{\tau}^{\boldsymbol{\pi}}} \left[\sum_{t=0}^{T-1} \| \boldsymbol{\sigma}_n(\boldsymbol{s}_t, \boldsymbol{\pi}(\boldsymbol{s}_t)) \| \right], \tag{3}$$

$$\boldsymbol{s}_{t+1} \sim \mathcal{N} \left(\boldsymbol{\mu}_n(\boldsymbol{s}_t, \boldsymbol{\pi}(\boldsymbol{s}_t)), \boldsymbol{\sigma}_n^2(\boldsymbol{s}_t, \boldsymbol{\pi}(\boldsymbol{s}_t)) \right) + \boldsymbol{w}_t. \tag{4}$$

Similar to domain randomization, by sampling proportional to the model uncertainty in Equation (4), we make our policy robust towards model inaccuracies while also optimizing for the active objective in Equation (3). We call this algorithm PETS-AE. Finally, Curi *et al.* show that trajectory sampling

Algorithm 1 SOFTAE

Init: Aleatoric uncertainty σ, Statistical model $(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0, \beta_0)$, $\mathcal{D}_0 := \emptyset$
for episode $n = 1, \ldots, N$ **do**

$\qquad \boldsymbol{\pi}_n = \text{OPTIMIZEPOLICY}(\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n, \beta_n)$ ➤ Prepare policy via Equations (5–6)

$\qquad (\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{S}') \leftarrow \text{ROLLOUT}(\boldsymbol{\pi}_n)$ ➤ Collect measurements

$\qquad \mathcal{D}_n \leftarrow \mathcal{D}_{n-1} \cup (\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{S}')$ ➤ Add rollout to dataset

$\qquad$ Update $(\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n, \beta_n, \mathcal{D}_n)$ ➤ Update model via Equation (7)

end for

---

often leads to the policy acting greedily with respect to the current model posterior and underperforms in practice [55]. We adapt their optimistic planner like so:

$$\boldsymbol{\pi}_n^* = \underset{\boldsymbol{\pi} \in \Pi, \boldsymbol{\eta} \in [-\beta, \beta]^{d_v s}}{\arg \max} J_n(\boldsymbol{\pi}) = \underset{\boldsymbol{\pi} \in \Pi, \boldsymbol{\eta} \in [-\beta, \beta]^{d_v s}}{\arg \max} \mathbb{E}_{\boldsymbol{\tau}^{\pi}} \left[\sum_{t=0}^{T-1} \|\boldsymbol{\sigma}_n(\boldsymbol{s}_t, \boldsymbol{\pi}(\boldsymbol{s}_t))\| \right], \qquad (5)$$

$$\boldsymbol{s}_{t+1} = \boldsymbol{\mu}_{n-1}(\boldsymbol{s}_t, \boldsymbol{\pi}(\boldsymbol{s}_t)) + \boldsymbol{\sigma}_{n-1}(\boldsymbol{s}_t, \boldsymbol{\pi}(\boldsymbol{s}_t))\boldsymbol{\eta}(\boldsymbol{s}_t) + \boldsymbol{w}_t. \qquad (6)$$

The hallucinated controls $\boldsymbol{\eta}$ in Equation (6) are used to optimize over the dynamics that lie in the set $\mathcal{M}(\boldsymbol{s}_t) = [\boldsymbol{\mu}_{n-1}(\boldsymbol{s}_t, \boldsymbol{a}_t) \pm \beta_{n-1}\boldsymbol{\sigma}_{n-1}(\boldsymbol{s}_t, \boldsymbol{a}_t)]$. Here β_{n-1} is treated as a hyperparameter. Hence, by maximizing over $\boldsymbol{\eta}$, we pick dynamics in the set $\mathcal{M}(\boldsymbol{s}_t)$ that are the most favorable for our objective $J_n(\boldsymbol{\pi})$ in Equation (5). We call this algorithm SOFTAE, as summarized in Algorithm 1.

Sukhija *et al.* propose the planning problem in Equations (5) and (6) and show that under common regularity assumptions on $\boldsymbol{f}^*$, the objective guarantees polynomial sample complexity [17], i.e., that we converge to an ϵ-optimal solution in polynomial time. This in turn implies that the estimate $\boldsymbol{\mu}_n$ converges to the true system $\boldsymbol{f}^*$ for $N \to \infty$. Recently, Sukhija *et al.* show that convergence also holds for both MEAN-AE and PETS-AE [61]. However, while all the aforementioned planning strategies guarantee convergence in theory, in our experiments, we use SOFTAE, since we found it to perform the best across tasks empirically. We report the comparisons between different planning strategies in fig. S1.

Learning Uncertainty Aware Dynamics

We use an ensemble of neural networks (NN) [56] to learn an uncertainty-aware state space model. We approximate the posterior distribution with a set of L NN parameter particles $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L\}$. The particles are sampled i.i.d. from a parameter distribution $\boldsymbol{\theta}_l^0 \sim p(\boldsymbol{\theta})$ and updated via maximum-likelihood estimation (MLE):

$$\boldsymbol{\theta}_l^{i+1} \leftarrow \boldsymbol{\theta}_l^i + \underbrace{\gamma}_{\text{Learning Rate}} \underbrace{\nabla_{\boldsymbol{\theta}_j^i} \ln p(\boldsymbol{S}'|\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{\theta}_l^i)}_{\text{MLE}}, \forall l \in \{1, \ldots, L\}. \qquad (7)$$

We pick $p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\theta}_l^i) := \mathcal{N}(\text{NN}(\boldsymbol{s}, \boldsymbol{a}|\boldsymbol{\theta}_l^i), \epsilon^2)$, resulting in the squared error loss for each particle. Given the particles $\{\boldsymbol{\theta}_1^n, \ldots, \boldsymbol{\theta}_L^n\}$ at iteration n, we estimate the mean and model uncertainty with

$$\boldsymbol{\mu}_n(\boldsymbol{s}, \boldsymbol{a}) \approx \frac{1}{L} \sum_{i=1}^{L} \text{NN}(\boldsymbol{s}, \boldsymbol{a}|\boldsymbol{\theta}_i^n), \quad \boldsymbol{\sigma}_n^2(\boldsymbol{s}, \boldsymbol{a}) \approx \text{Var}\left(\{\text{NN}(\boldsymbol{s}, \boldsymbol{a}|\boldsymbol{\theta}_i^n)\}_{i=1}^{L}\right) \qquad (8)$$

15

Intuitively, as we collect more data in our domain $\mathcal{S} \times \mathcal{A}$, we expect the different particle initializations to converge to the same prediction and therefore our epistemic uncertainty to decrease. This holds for classical models such as Gaussian processes [66] and has also been shown empirically for NN models [69].

Trajectory and Policy Optimization

Equations (1–6) describe the reward function and the transition dynamics we use for optimizing the policy. In the following, we discuss two approaches for obtaining a closed-loop policy for the data collection.
Trajectory Optimization and Model-Predictive Control Instead of learning an explicit policy function, here, we directly optimize over the actions $\{\boldsymbol{a}_0, \boldsymbol{a}_1, \dots\}$ for a fixed horizon $H < T$ and apply receding horizon/model predictive control [70]. For instance, instead of solving for the policy in Equation (1), we solve the following trajectory optimization problem:

$$\{\boldsymbol{a}_0^*, \boldsymbol{a}_1^*, \dots, \boldsymbol{a}_H^*\} = \underset{\boldsymbol{a}_0, \boldsymbol{a}_1, \dots, \boldsymbol{a}_H \in \mathcal{A}^H}{\arg\max} \; J_n(\{\boldsymbol{a}_0, \boldsymbol{a}_1, \dots, \boldsymbol{a}_H\}) \tag{9}$$

$$= \underset{\boldsymbol{a}_0, \boldsymbol{a}_1, \dots, \boldsymbol{a}_H \in \mathcal{A}^H}{\arg\max} \; \mathbb{E}_{\boldsymbol{\tau}^{\boldsymbol{\pi}}, \hat{\boldsymbol{s}}_0 = \boldsymbol{s}_t} \left[\sum_{h=0}^{H-1} \|\boldsymbol{\sigma}_n(\hat{\boldsymbol{s}}_h, \boldsymbol{a}_h)\| \right],$$

$$\hat{\boldsymbol{s}}_{h+1} = \boldsymbol{\mu}_n(\hat{\boldsymbol{s}}_h, \boldsymbol{a}_h) + \boldsymbol{w}_t.$$

Next, we apply the first control $\boldsymbol{a}_0^*$ to the real system and observe the next state $\boldsymbol{s}_{t+1}$. We repeat the optimization above with $\hat{\boldsymbol{s}}_0 = \boldsymbol{s}_{t+1}$ to obtain the next control input. By repeating the optimization at each timestep, we obtain an implicit closed-loop controller. This approach does not require a policy parameterization, is simple, and is often more stable during learning. We use a sampling-based solver, in particular [57], for Equation (9). Here, we use a candidate distribution over the action sequence $p(\{\boldsymbol{a}_0, \boldsymbol{a}_1, \dots, \boldsymbol{a}_H\})$, e.g., a clipped Gaussian, and then generate P samples $\{\boldsymbol{a}_0^p, \boldsymbol{a}_1^p, \dots, \boldsymbol{a}_H^p\}_{p=0}^P$. We evaluate the objective for all the samples $J_n(\{\boldsymbol{a}_0^p, \boldsymbol{a}_1^p, \dots, \boldsymbol{a}_H^p\})$ and use its value to update the sampling distribution $p(\{\boldsymbol{a}_0, \boldsymbol{a}_1, \dots, \boldsymbol{a}_H\})$. This procedure is repeated over several steps, following which the best candidate is returned (see [57] for more details).
Model-Based Policy Optimization As an alternative to trajectory optimization, we can also learn a parametrized policy $\boldsymbol{\pi}_\phi$ directly using the learned dynamics. Concretely, we sample a batch of P states $\{\boldsymbol{s}_i\}_{i=1}^P$ from a replay buffer of real state transitions gathered so far and train the policy with Soft Actor-Critic (SAC, c.f. [71]), which combines off-policy learning with entropy maximization to balance exploitation and exploration. We augment the limited real-world data with synthetic rollouts of length H generated from the learned dynamics. For each sampled state, we simulate

$$\hat{\boldsymbol{s}}_{h+1} = \boldsymbol{\mu}_n(\hat{\boldsymbol{s}}_h, \boldsymbol{a}_h) + \boldsymbol{w}_h, \quad \boldsymbol{a}_h \sim \boldsymbol{\pi}_\phi(\cdot|\hat{\boldsymbol{s}}_h), \quad h = 0, \dots, H-1, \tag{10}$$

with $\hat{\boldsymbol{s}}_0 = \boldsymbol{s}_i$. The resulting imagined trajectories $(\hat{\boldsymbol{s}}_h, \boldsymbol{a}_h, \hat{\boldsymbol{s}}_{h+1})$ branched from real data are then combined with real transitions to form the training set for policy optimization, which improves sample efficiency [64]. To further stabilize learning, we apply the symlog transform [63] to observations in order to reduce sensitivity to outliers. Specifically, for a scalar input x we train the dynamics model on

$$\mathrm{symlog}(x) = \mathrm{sign}(x) \log\left(1 + |x|\right), \tag{11}$$

which preserves the sign and compresses large magnitudes while remaining approximately linear near zero. Its inverse for mapping back to the original scale is

$$\mathrm{symexp}(y) = \mathrm{sign}(y) \left(e^{|y|} - 1\right), \tag{12}$$

16

which yields bounded, well-behaved gradients and helps prevent rare spikes in sensor values from destabilizing training.

Unlike trajectory optimization, which repeatedly solves an open-loop action sequence optimization at each time step, model-based policy optimization amortizes this computation into the policy parameters ϕ. This yields a closed-loop controller that is both sample-efficient—thanks to model rollouts—and computationally efficient at deployment. However, a potential drawback is that inaccuracies in the learned dynamics can mislead the policy, especially early on. If π_ϕ then exploits model errors, this bias can propagate through imagined rollouts and destabilize subsequent learning.

Soft Robotic Simulated Environments and Tasks

Soft Continuum Arm via Cosserat Rod Simulation

The simulated soft robotic arm environment is grounded in Cosserat rod theory, implemented using the Elastica simulator [72]. The system models a slender continuous soft arm that exhibits nonlinear bending deformation. Actuation is applied as internal torques in the normal and binormal planes, controlled via 6 evenly spaced control points along the arm. This results in a total of 12 actuation degrees of freedom. The system's 58-dimensional state space includes spatially distributed representations of position, linear velocity, orientation, and angular velocity along the length of the continuously deformable arm (Figure 1A), making the dynamics both high-dimensional and highly nonlinear.

We evaluate the learned dynamics models on two endpoint reaching tasks that differ in difficulty. In both cases, the goal is to control the soft arm to reach a fixed target position within its workspace. Task *(i)* places the target with a small distance away from the rest state, requiring only moderate deformation and resulting in relatively smooth, low-curvature trajectories. The second task *(ii)* places the target closer to the edge of the workspace, demanding more extreme bending and invoking stronger nonlinear dynamics.

Articulated Fish with Deformable Skin in Fluid

The second simulated system is a soft robotic fish in a fluid environment, implemented using FishGym [73], a high-performance simulator with two-way coupled fluid-structure interaction. As illustrated in Figure 1B, the robot consists of an articulated skeleton enclosed in a deformable skin mesh and is immersed in a simulated viscous fluid. This setup captures complex hydrodynamic effects such as vortex shedding, body–fluid coupling, and delayed actuation responses, making it an extremely challenging environment for dynamics learning.

The fish is actuated through four internal joints along its spine, each controlled by a scalar torque input, enabling flexible body undulation for thrust generation. To assess the generalization of the learned dynamics model, we define four aquatic locomotion tasks: *(iii)* swim forward along the +x direction, *(iv)* execute a U-turn and swim along the –x direction, *(v)* reach a randomized target location, and *(vi)* swim away from a randomized target. For all tasks, we constrain the robot fish to swim on a horizontal 2D plane without buoyancy control. Tasks (iii) and (iv) use a 15-dimensional state space comprising the fish's velocity, orientation, and joint positions and velocities. Tasks (v) and (vi) extend this with an additional 3-dimensional vector representing the relative target position to the fish.

Musculoskeletal Leg with Electrohydraulic Muscles

Inspired by recent advances in electrohydraulic musculoskeletal legs [74], the third simulated environment models a planar robotic leg capable of fast, adaptive motion with tunable stiffness and high energy efficiency. The leg is rigidly attached to a circular boom that rotates around a fixed vertical axis, constraining the system to move along a horizontal arc. Actuation is provided at the hip and knee joints via antagonistic pairs of contracting electrohydraulic artificial muscles, enabling compliant, muscle-driven dynamics.

To simulate this system, we implement the musculoskeletal (MSK) leg in MuJoCo (Figure 1C). The soft muscle actuators are modeled using a neural network trained on real electrohydraulic actuator data, capturing nonlinear voltage-to-force characteristics. An additional DC motor added at the boom joint allows us to control the boom's roll angle, which effectively adjusts the pitch and hopping direction of the leg. This motor enables hopping behaviors in different directions on a circular track.

The state space is 10-dimensional and includes the yaw, pitch, and roll angles of the boom, the hip and knee joint angles, and the corresponding angular velocities for each of these five degrees of freedom. The action space comprises 4 muscle voltage commands and the position control signal to the DC motor. We define two locomotion tasks for this system: *(vii)* hop counterclockwise and *(viii)* hop clockwise along the circular trajectory. These tasks require coordinated actuation between the compliant leg and boom orientation, highlighting the challenge of learning hybrid, muscle-driven dynamics.

References and Notes

[1] Oncay Yasa, Yasunori Toshimitsu, Mike Y Michelis, Lewis S Jones, Miriam Filippi, Thomas Buchner, and Robert K Katzschmann. An overview of soft robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 6(1):1–29, 2023.

[2] Cosimo Della Santina, Christian Duriez, and Daniela Rus. Model-based control of soft robots: A survey of the state of the art and open challenges. *IEEE Control Systems Magazine*, 43(3):30–65, 2023.

[3] Tao Du, Josie Hughes, Sebastien Wah, Wojciech Matusik, and Daniela Rus. Underwater soft robot modeling and control with differentiable simulation. *IEEE Robotics and Automation Letters*, 6:4994–5001, 2021.

[4] Jue Wang and Alex Chortos. Control strategies for soft robot systems. *Advanced Intelligent Systems*, 4:2100165, 2022.

[5] Wu-Te Yang, Hannah S Stuart, Burak Kürkçü, and Masayoshi Tomizuka. Nonlinear modeling for soft pneumatic actuators via data-driven parameter estimation. In *2024 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 642–648. IEEE, 2024.

[6] Longhui Qin, Haijun Peng, Xiaonan Huang, Mingchao Liu, and Weicheng Huang. Modeling and simulation of dynamics in soft robotics: A review of numerical approaches. *Current Robotics Reports*, 5:1–13, 2024.

[7] Rogelio Ortigosa-Martínez, Jesús Martínez-Frutos, Carlos Mora-Corral, Pablo Pedregal, and Francisco Periago. Mathematical modeling, analysis and control in soft robotics: a survey. *SeMA Journal*, 81:147–164, 2024.

[8] Shengkai Liu, Hongfei Yu, Ning Ding, Xuchun He, Hengli Liu, and Jun Zhang. Exploring modeling techniques for soft arms: A survey on numerical, analytical, and data-driven approaches. *Biomimetics*,

10:71, 2025.

[9] Filippo A. Spinelli and Robert K. Katzschmann. A unified and modular model predictive control framework for soft continuum manipulators under internal and external constraints. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9393–9400. IEEE, 2022.

[10] Amirhossein Kazemipour, Oliver Fischer, Yasunori Toshimitsu, Ki Wan Wong, and Robert K. Katzschmann. Adaptive dynamic sliding mode control of soft continuum manipulators. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3259–3265. IEEE, 2022.

[11] Oliver Fischer, Yasunori Toshimitsu, Amirhossein Kazemipour, and Robert K. Katzschmann. Dynamic task space control enables soft manipulators to perform real-world tasks. *Advanced Intelligent Systems*, 5:2200024, 2023.

[12] Jian Qu, Kamran Khan, Songhe Meng, and Anastasia Muliana. Modeling nonlinear viscoelastic responses of flexible composites for soft robotics applications. *Mechanics of Advanced Materials and Structures*, 30(14):2793–2805, 2023.

[13] Maziar Raissi, Paris Perdikaris, and George 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*, 378:686–707, 2019.

[14] 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*, 9:8523–8530, 2024.

[15] Chenhao Li, Andreas Krause, and Marco Hutter. Robotic world model: A neural network simulator for robust policy optimization in robotics. 2025. arXiv:2501.10100.

[16] Cansu Sancaktar, Sebastian Blaes, and Georg Martius. Curious exploration via structured world models yields zero-shot object manipulation. *Advances in Neural Information Processing Systems*, 35:24170–24183, 2022.

[17] Bhavya Sukhija, Lenart Treven, Cansu Sancaktar, Sebastian Blaes, Stelian Coros, and Andreas Krause. Optimistic active exploration of dynamical systems. *Advances in Neural Information Processing Systems*, 36:38122–38153, 2023.

[18] Chenhao Li, Andreas Krause, and Marco Hutter. Offline robotic world model: Learning robotic policies without a physics simulator. 2025. arXiv:2504.16680.

[19] Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, 40(4-5):698–721, 2021.

[20] Mathieu Dubied, Mike Yan Michelis, Andrew Spielberg, and Robert Kevin Katzschmann. Sim-to-real for soft robots using differentiable fem: Recipes for meshing, damping, and actuation. *IEEE Robotics and Automation Letters*, 7(2):5015–5022, 2022.

[21] Anup Teejo Mathew, Ikhlas Ben Hmida, Costanza Armanini, Frederic Boyer, and Federico Renda. Sorosim: A matlab toolbox for hybrid rigid-soft robots based on the geometric variable-strain approach. *IEEE Robotics & Automation Magazine*, 30:106–122, 2022.

[22] Yasunori Toshimitsu, Ki Wan Wong, Thomas Buchner, and Robert Katzschmann. Sopra: Fabrication & dynamical modeling of a scalable soft continuum robotic arm with integrated proprioceptive sensing. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 653–660. IEEE, 2021.

[23] Hehui Zheng, Sebastian Pinzello, Barnabas Gavin Cangan, Thomas J K Buchner, and Robert K

Katzschmann. Vision-based online key point estimation of deformable robots. *Advanced Intelligent Systems*, 6:2400105, 2024.

[24] Panagiotis Polygerinos, Zheng Wang, Johannes TB Overvelde, Kevin C Galloway, Robert J Wood, Katia Bertoldi, and Conor J Walsh. Modeling of soft fiber-reinforced bending actuators. *IEEE Transactions on Robotics*, 31:778–789, 2015.

[25] Shota Kokubu, Pablo E Tortós Vinocour, and Wenwei Yu. Development and evaluation of fiber reinforced modular soft actuators and an individualized soft rehabilitation glove. *Robotics and Autonomous Systems*, 171:104571, 2024.

[26] Xinda Qi, Yu Mei, Dong Chen, Zhaojian Li, and Xiaobo Tan. Design and nonlinear modeling of a modular cable-driven soft robotic arm. *IEEE/ASME Transactions on Mechatronics*, 29:3083–3091, 2024.

[27] Barnabas Gavin Cangan, Stefan Escaida Navarro, Bai Yang, Yu Zhang, Christian Duriez, and Robert K Katzschmann. Model-based disturbance estimation for a fiber-reinforced soft manipulator using orientation sensing. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9424–9430. IEEE, 2022.

[28] Pasquale Ferrentino, Antonio López-Díaz, Seppe Terryn, Julie Legrand, Joost Brancart, Guy Van Assche, Ester Vázquez, Andrés Vázquez, and Bram Vanderborght. Quasi-static fea model for a multi-material soft pneumatic actuator in sofa. *IEEE Robotics and Automation Letters*, 7:7391–7398, 2022.

[29] Hao Chen, Jian Chen, Xinran Liu, Zihui Zhang, Yuanrui Huang, Zhongkai Zhang, and Hongbin Liu. Accelerated quasi-static fem for real-time modeling of continuum robots with multiple contacts and large deformation. 2025. arXiv:2101.12345.

[30] Robert K Katzschmann, Maxime Thieffry, Olivier Goury, Alexandre Kruszewski, Thierry-Marie Guerra, Christian Duriez, and Daniela Rus. Dynamically closed-loop controlled soft robotic arm using a reduced order finite element model with state observer. In *2019 IEEE International Conference on Soft Robotics (RoboSoft)*, pages 717–724. IEEE, 2019.

[31] Sander Tonkens, Joseph Lorenzetti, and Marco Pavone. Soft robot optimal control via reduced order finite element models. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12010–12016. IEEE, 2021.

[32] Francesco Stella, Qinghua Guan, Cosimo Della Santina, and Josie Hughes. Piecewise affine curvature model: a reduced-order model for soft robot-environment interaction beyond pcc. In *2023 IEEE International Conference on Soft Robotics (RoboSoft)*, pages 1–7. IEEE, 2023.

[33] Thomas George Thuruthel, Benjamin Shih, Cecilia Laschi, and Michael Thomas Tolley. Soft robot perception using embedded soft sensors and recurrent neural networks. *Science Robotics*, 4:eaav1488, 2019.

[34] Thomas George Thuruthel, Egidio Falotico, Federico Renda, and Cecilia Laschi. Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators. *IEEE Transactions on Robotics*, 35:124–134, 2019.

[35] Morgan T. Gillespie, Charles M. Best, Eric C. Townsend, David Wingate, and Marc D. Killpack. Learning nonlinear dynamic models of soft robots for model predictive control with neural networks. In *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, pages 39–45, 2018.

[36] Audrey Sedal, Alan Wineman, R Brent Gillespie, and C David Remy. Comparison and experimental validation of predictive models for soft, fiber-reinforced actuators. *The International Journal of*

Robotics Research, 40:119–135, 2021.

[37] Zixi Chen, Federico Renda, Alexia Le Gall, Lorenzo Mocellin, Matteo Bernabei, Théo Dangel, Gastone Ciuti, Matteo Cianchetti, and Cesare Stefanini. Data-driven methods applied to soft robot modeling and control: A review. *IEEE Transactions on Automation Science and Engineering*, 22:2241–2256, 2025.

[38] Yuxiang Yang, Ken Caluwaerts, Atil Iscen, Tingnan Zhang, Jie Tan, and Vikas Sindhwani. Data efficient reinforcement learning for legged robots. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 1–10. PMLR, 2020.

[39] Suyoung Choi, Gwanghyeon Ji, Jeongsoo Park, Hyeongjun Kim, Juhyeok Mun, Jeong Hyun Lee, and Jemin Hwangbo. Learning quadrupedal locomotion on deformable terrain. *Science Robotics*, 8:eade2256, 2023.

[40] Chenhao Li, Elijah Stanger-Jones, Steve Heim, and Sang bae Kim. FLD: Fourier latent dynamics for structured motion representation and learning. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2024.

[41] Jacob Levy, Tyler Westenbroek, and David Fridovich-Keil. Learning to walk from three minutes of real-world data with semi-structured dynamics models. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 2061–2079. PMLR, 2025.

[42] Jonas Rothfuss, Bhavya Sukhija, Lenart Treven, Florian Dörfler, Stelian Coros, and Andreas Krause. Bridging the sim-to-real gap with bayesian inference. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10784–10791. IEEE, 2024.

[43] Todd Hester and Peter Stone. Intrinsically motivated model learning for developing curious robots. *Artificial Intelligence*, 247:170–186, 2017.

[44] Arthur Aubret, Laetitia Matignon, and Salima Hassas. A survey on intrinsic motivation in reinforcement learning. 2019. arXiv:1908.06976.

[45] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8583–8592. PMLR, 2020.

[46] Artem Latyshev and Aleksandr I. Panov. Intrinsic motivation in model-based reinforcement learning: A brief review. *Scientific and Technical Information Processing*, 51:460–470, 2024.

[47] Taekyung Kim, Jungwi Mun, Junwon Seo, Beomsu Kim, and Seongil Hong. Bridging active exploration and uncertainty-aware deployment using probabilistic ensemble neural network dynamics. In *Proceedings of Robotics: Science and Systems*, 2023.

[48] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2778–2787. PMLR, 2017.

[49] Sarah Bechtle, Yixin Lin, Akshara Rai, Ludovic Righetti, and Franziska Meier. Curious ilqr: Resolving uncertainty in model-based rl. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 162–171. PMLR, 2020.

[50] Tim Seyde, Wilko Schwarting, Sertac Karaman, and Daniela Rus. Learning to plan optimistically: Uncertainty-guided deep exploration via latent model ensembles. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 1156–1167. PMLR, 2022.

[51] Urban Fasel, J Nathan Kutz, Bingni W Brunton, and Steven L Brunton. Ensemble-sindy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proceedings of the Royal Society A*, 478:20210904, 2022.

[52] Xiangyu Shao, Linke Xu, Guanghui Sun, Weiran Yao, Ligang Wu, and Cosimo Della Santina. Self-attention enhanced dynamics learning and adaptive fractional-order control for continuum soft robots with system uncertainties. *IEEE Transactions on Automation Science and Engineering*, 22:18694–18708, 2025.

[53] Xiao Liu, Shuhei Ikemoto, Yuhei Yoshimitsu, and Heni Ben Amor. Learning soft robot dynamics using differentiable kalman filters and spatio-temporal embeddings. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2550–2557. IEEE, 2023.

[54] Rianna Jitosho, Tyler Ga Wei Lum, Allison Okamura, and Karen Liu. Reinforcement learning enables real-time planning and control of agile maneuvers for soft robot arms. 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 1131–1153. PMLR, 2023.

[55] Sebastian Curi, Felix Berkenkamp, and Andreas Krause. Efficient model-based reinforcement learning through optimistic policy search and planning. *Advances in Neural Information Processing Systems*, 33:14156–14170, 2020.

[56] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30:6405—-6416, 2017.

[57] Cristina Pinneri, Shambhuraj Sawant, Sebastian Blaes, Jan Achterhold, Joerg Stueckler, Michal Rolinek, and Georg Martius. Sample-efficient cross-entropy method for real-time planning. In *Proceedings of the Conference on Robot Learning*, pages 1049–1065. PMLR, 2021.

[58] Materials and methods are available as supplementary material.

[59] Yahia A AboZaid, Mahmoud T Aboelrayat, Irene S Fahim, and Ahmed G Radwan. Soft robotic grippers: A review on technologies, materials, and applications. *Sensors and Actuators A: Physical*, page 115380, 2024.

[60] Linda Paternò and Lucrezia Lorenzon. Soft robotics in wearable and implantable medical applications: Translational challenges and future outlooks. *Frontiers in Robotics and AI*, 10:1075634, 2023.

[61] Bhavya Sukhija, Lenart Treven, Carmelo Sferrazza, Florian Dorfler, Pieter Abbeel, and Andreas Krause. Sombrl: Scalable and optimistic model-based rl. *Advances in Neural Information Processing Systems*, 38, 2025.

[62] Bhavya Sukhija, Lenart Treven, Florian Dorfler, Stelian Coros, and Andreas Krause. Neorl: Efficient exploration for nonepisodic rl. *Advances in Neural Information Processing Systems*, 37:74966–74998, 2024.

[63] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. 2019. arXiv:2101.12345.

[64] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-

based policy optimization. *Advances in Neural Information Processing Systems*, 32:12519–12530, 2019.

[65] Klemens Iten, Lenart Treven, Bhavya Sukhija, Florian Dörfler, and Andreas Krause. Sample-efficient and scalable exploration in continuous-time rl, 2025.

[66] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[67] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in Neural Information Processing Systems*, 31:4759–4770, 2018.

[68] Mona Buisson-Fenet, Friedrich Solowjow, and Sebastian Trimpe. Actively learning gaussian process dynamics. In Alexandre M. Bayen, Ali Jadbabaie, George Pappas, Pablo A. Parrilo, Benjamin Recht, Claire Tomlin, and Melanie Zeilinger, editors, *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 5–15. PMLR, 2020.

[69] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5062–5071. PMLR, 2019.

[70] Manfred Morari and Jay H Lee. Model predictive control: Past, present and future. *Computers & Chemical Engineering*, 23:667–682, 1999.

[71] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR, 2018.

[72] Noel Naughton, Jiarui Sun, Arman Tekinalp, Tejaswin Parthasarathy, Girish Chowdhary, and Mattia Gazzola. Elastica: A compliant mechanics environment for soft robotic control. *IEEE Robotics and Automation Letters*, 6:3389–3396, 2021.

[73] Wenji Liu, Kai Bai, Xuming He, Shuran Song, Changxi Zheng, and Xiaopei Liu. Fishgym: A high-performance physics-based simulation framework for underwater robot learning. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6268–6275. IEEE, 2022.

[74] Thomas J K Buchner, Toshihiko Fukushima, Amirhossein Kazemipour, Stephan-Daniel Gravert, Manon Prairie, Pascal Romanescu, Philip Arm, Yu Zhang, Xingrui Wang, Steven L Zhang, Johannes Walter, Christoph Keplinger, and Robert K. Katzschmann. Electrohydraulic musculoskeletal robotic leg for agile, adaptive, yet energy-efficient locomotion. *Nature Communications*, 15:7634, 2024.

[75] Max Simchowitz and Dylan Foster. Naive exploration is optimal for online lqr. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8937–8948. PMLR, 2020.

[76] Bhavya Sukhija, Stelian Coros, Andreas Krause, Pieter Abbeel, and Carmelo Sferrazza. Maxinforl: Boosting exploration in reinforcement learning through information gain maximization. In *The Thirteenth International Conference on Learning Representations*, 2025.

Supplementary Materials and Methods

Comparison of Different Uncertainty-Driven Active Exploration Strategies

In the main paper, we focus on SOFTAE, our optimistic active exploration strategy for data-efficient and task-agnostic dynamics learning in soft robotic systems. Here, we compare SOFTAE with two additional uncertainty-driven active exploration baselines, MEAN-AE and PETS-AE, alongside RANDOM exploration and the task-specific model-based method H-UCRL.

All active exploration methods use probabilistic ensembles to model the system dynamics, providing both a mean prediction $\boldsymbol{\mu}_n$ and an uncertainty estimate $\boldsymbol{\sigma}_n$ of the next state at each state-action pair. The exploration policy $\boldsymbol{\pi}_n$ is selected to maximize predicted model uncertainty over the rollout horizon, thereby driving the agent toward less-explored regions of the state-action space. MEAN-AE plans trajectories using the mean model $\boldsymbol{\mu}_n$ directly. While straightforward, this approach can lead to biased state estimate, as even small model errors can accumulate quickly over time [67]. It is also proven suboptimal outside of linear systems [75], making the mean estimator less suitable for soft robotic systems, which are inherently nonlinear and underactuated. PETS-AE follows the probabilistic ensemble trajectory sampling approach, sampling from $\mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2)$ during planning to increase robustness to model inaccuracies. SOFTAE uses optimistic planning by augmenting the mean model with *hallucinated controls* that explore the most favourable dynamics within the model's uncertainty set.

Although all three active exploration methods are theoretically guaranteed to converge to the true dynamics under mild assumptions, we show in Figure S1 that SOFTAE achieves the most consistent and highest returns across tasks. In particular, SOFTAE demonstrates clear advantages on challenging generalization scenarios such as *(ii)* reach far, *(iv)* turn & swim, and *(vi)* swim away. Returns are averaged over 10 seeds for tasks *(i)* and *(ii)*, and 5 seeds for the remaining tasks, shaded regions indicate standard deviation.

SOFTAE with Model-Based Policy Optimization

To investigate the integration of SOFTAE with model-based policy optimization (MBPO) for obtaining closed-loop policies, we implement a proof-of-concept variant on the soft continuum arm environment. In all cases, policies are trained with Soft Actor-Critic (SAC, c.f. [71]) and we use the implementation from [76]. Training uses sampled collected states augmented with simulated rollouts from the learned dynamics model, following standard MBPO protocols [63–65].

We compare our SOFTAE-MBPO against three baselines: RANDOM, which collects data from uncontrolled actuation without exploration or task guidance; and two standard task-oriented model-based policy optimization baselines, MBPO (using the mean model for simulated rollouts) and MBPO-TS (using an uncertainty-based trajectory sampling scheme).

As shown in Fig. S2, SOFTAE-MBPO achieves performance on par with MBPO and MBPO-TS on their training task *(i)*, but substantially outperforms them on previously unseen downstream tasks. In contrast, the RANDOM baseline lags behind across all tasks, reflecting the inefficiency of unguided data collection. The superior generalization of SOFTAE-MBPO arises from its active exploration strategy, which prioritizes underexplored regions rather than overfitting to task-specific rewards. A known limitation, however, is the instability of policy optimization, since inaccuracies in the learned dynamics may mislead the policy early in training. By contrast, trajectory optimization used in our main experiments re-optimizes open-loop action sequences at each time step, making it less sensitive to such bias. Addressing this instability would require frequent retraining of the policy after each model update, which is computationally more expensive.

Environment Details

We provide detailed descriptions of the state and action spaces for the three simulated soft robotic environments and one real-world setup used in our experiments: the soft continuum arm, the deformable fish in fluid, the musculoskeletal (MSK) leg, and the SoPrA soft continuum arm. These systems differ in morphology, actuation, and task structure, resulting in varying observation state and control action dimensionalities.

Table S1 outlines the state and action decomposition for the simulated soft continuum arm, which is discretized into either 5 or 11 elements along its body, yielding 58D and 136D state spaces, respectively. The state includes spatially distributed position, linear velocity, orientation, and angular velocity entries, while the action space consists of torques applied in the normal and binormal directions at 6 control points.

Table S2 summarizes the state and action for the articulated fish with deformable skin, as used in tasks *(iii)* to *(vi)*. The 15D state space applies to the simpler forward and U-turn swimming tasks *(iii)* and *(iv)*, while the 18D version includes an additional target-relative position vector for goal-reaching or avoiding tasks *(v)* and *(vi)*. The fish is actuated by four internal joint torques.

Table S3 describes the musculoskeletal (MSK) leg environment, which is controlled via a combination of a DC motor at the boom joint and four voltage inputs to antagonistic electrohydraulic muscles. The state space captures the boom's full orientation and angular velocity, along with the angles and velocities of the leg's two joints.

Finally, Table S4 presents the real-world soft continuum SoPrA arm environment. The system is tracked using a motion capture software (Qualisys Track Manager) and actuated by six differential pressure commands, each applied to one of the arm's air chambers. The state space consists of the position, linear velocity, orientation, and angular velocity of both the midpoint and tip motion-capture rings relative to the arm base ring (Figure 4), along with the six previous step control commands, represented as normalized absolute pressures.

Table S5 details the reward functions used for each downstream task across the different environments, designed to align with the specific objectives of each task.

Experiment Details

The hyperparameters used for our experiments are presented here. We train the dynamics model after each episode of data collection. For training, we fix the number of epochs to determine the number of gradient steps. The hyperparameters for dynamics model training, iCEM optimizer, and SAC policy training are presented in Tables S6 to S8, respectively. The SAC hyperparameters correspond to the model-based policy training experiments on the simulated soft continuum arm, as reported in Fig. S2.
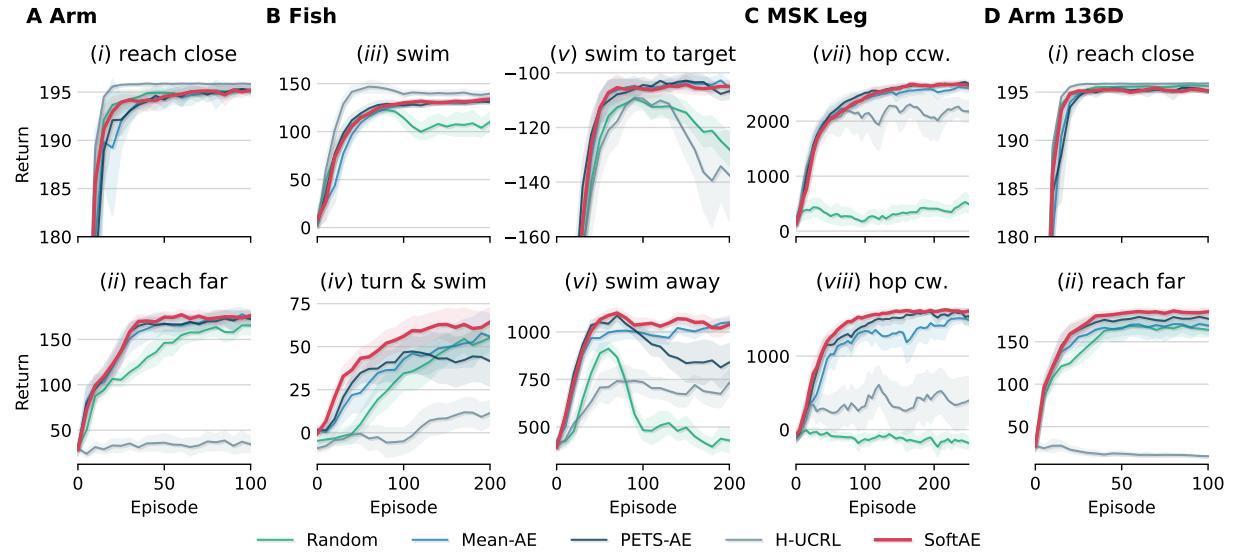
A Arm

(*i*) reach close

(*ii*) reach far

B Fish

(*iii*) swim

(*iv*) turn & swim

(*v*) swim to target

(*vi*) swim away

C MSK Leg

(*vii*) hop ccw.

(*viii*) hop cw.

D Arm 136D

(*i*) reach close

(*ii*) reach far

Random — Mean-AE — PETS-AE — H-UCRL — SoftAE

Figure S1: **Comparison of different uncertainty-driven active exploration strategies.** Same as return plots in Figure 2 but with two additional active exploration baselines: Mean-AE and PETS-AE. Mean-AE plans trajectories using the mean model $\boldsymbol{\mu}_n$, PETS-AE uses trajectory sampling from the model posterior to improve robustness to model inaccuracies.
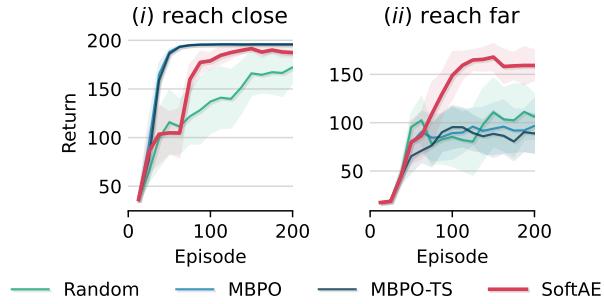
Figure S2: **Return comparison of policy optimization in the soft continuum arm simulation.** All methods use a learned dynamics model to train policies with SAC. Returns are averaged over 10 random seeds. SOFTAE+MBRL employs uncertainty-based rewards for active exploration, while PETS and the mean baselines are trained on task *(i)*. The random baseline collects trajectories from random actuation. SOFTAE+MBRL achieves performance comparable to task-specific baselines on the training task, but substantially outperforms them on unseen downstream tasks, highlighting the benefit of active exploration for generalization.

Table S1: **State and action space decomposition of the soft continuum arm.** This table details the individual state and action entries corresponding to the two observation discretization levels described in Table 1. State entries include element positions, velocities, orientations, and angular velocities, while action entries correspond to applied torques at six equidistant points along the arm. Dimensions are provided for both the 5-observed-point (58D) and 11-observed-point (136D) configurations.

State Entry	Symbol	Dimensions (58D)	Dimensions (136D)
element positions	$\mathbf{p}$	0:15	0:33
element linear velocities	$\dot{\mathbf{p}}$	15:30	33:66
element orientations	$\mathbf{R}$	30:46	66:106
element angular velocities	$\boldsymbol{\omega}$	46:58	106:136

Action Entry	Symbol	Dimensions	
normal torques	$\boldsymbol{u}_n$	0:6	
binormal torques	$\boldsymbol{u}_b$	6:12	

Table S2: **State and action space decomposition of the articulated fish with deformable skin.** This table follows the same format as Table S1. Dimensions are listed for task *(iii–iv)* with a 15D state space and task *(v–vi)* with an 18D state space, respectively.

State Entry	Symbol	Dimensions (15D)	Dimensions (18D)
relative position of target	$\mathbf{p}_{\text{target}} - \mathbf{p}$	-	0:3
base linear velocity	$\mathbf{v}$	0:3	3:6
head orientation	θ	3	6
head angular velocity	ω	4	7
joint positions	$\mathbf{q}$	5:10	8:13
joint velocities	$\dot{\mathbf{q}}$	10:15	13:18

Action Entry	Symbol	Dimensions	
joint torques	$\boldsymbol{u}$	0:4	

Table S3: **State and action space decomposition for the musculoskeletal (MSK) leg with hybrid actuation.** This table follows the same format as Table S1.

State Entry	Symbol	Dimensions
boom yaw	ψ	0
boom pitch	θ	1
boom roll	ϕ	2
leg joint positions	$\mathbf{q}$	3:5
boom angular velocities	ω	5:8
leg joint velocities	$\dot{\mathbf{q}}$	8:10

Action Entry	Symbol	Dimensions
DC motor position target	u	0
muscle voltages	V	1:5

Table S4: **State and action space decomposition for the real-world pneumatically actuated soft arm (SoPrA).** This table follows the same format as Table S1.

State Entry	Symbol	Dimensions
tip capture ring position	$\mathbf{p}$	0:3
tip capture ring orientation	$\mathbf{R}$	3:7
midpoint capture ring position	$\mathbf{p}_{\mathrm{mid}}$	7:10
midpoint capture ring orientation	$\mathbf{R}_{\mathrm{mid}}$	10:14
tip capture ring linear velocity	$\dot{\mathbf{p}}$	14:17
tip capture ring angular velocity	$\boldsymbol{\omega}$	17:20
midpoint capture ring linear velocity	$\dot{\mathbf{p}}_{\mathrm{mid}}$	20:23
midpoint capture ring angular velocity	$\boldsymbol{\omega}_{\mathrm{mid}}$	23:26
previous commands (normalized absolute pressures)	$\tilde{P}_{\mathrm{prev}}$	26:32

Action Entry	Symbol	Dimensions
delta pressures	ΔP	0:6

Table S5: **Downstream task rewards for simulated and real-world environments.** This table summarizes the reward definitions for all simulated environments described in Table 1 as well as for the real-world SoPrA platform. Each task reward is formulated as a function of position, linear velocity, or angular displacement, and follows the notation introduced in Tables S1–S4, depending on the environment and objective.

Environment	Task	Reward r_t
Arm	*(i) - (ii)* reach target	$g(\|\mathbf{p}_t - \mathbf{p}_{\text{target}}\|_2)^*$
Fish	*(iii)* swim forward along +x direction	$v_{+x,t}$
	(iv) turn and swim along -x direction	$v_{-x,t}$
	(v) swim to target	$-\|\mathbf{p}_t - \mathbf{p}_{\text{target}}\|_2$
	(vi) swim away from target	$\|\mathbf{p}_t - \mathbf{p}_{\text{target}}\|_2$
MSK Leg	*(vii)* hop counterclockwise	$\dot{\phi}_t$
	(viii) hop clockwise	$-\dot{\phi}_t$
SoPrA	*(ix)-(x)* reach target	$g(\|\mathbf{p}_t - \mathbf{p}_{\text{target}}\|_2)^*$

---

*For arm reach tasks, we employ a shaped long-tail reward function that maps the distance to target into the range $(0, 1]$. Specifically, $g(\|\mathbf{p}_t - \mathbf{p}_{\text{target}}\|_2) = 1$ when $\|\mathbf{p}_t - \mathbf{p}_{\text{target}}\|_2 \leq 5 \times 10^{-3}$.

Table S6: **Hyperparameters used in the experiments.** This table lists the dynamics model training hyperparameters for all soft robotic environments. The reported values cover exploration and task horizons, dynamics model network architecture and ensemble size, and training schedule.

Hyperparameters	Arm	Fish	MSK Leg	SoPrA
Exploration horizon	200	200	500	200
Downstream task horizon	200	200	500	20
Hidden layers	4	4	4	4
Neurons per layers	256	256	256	256
Number of ensembles	5	5	5	5
Batch size	64	64	64	64
Learning rate	5×10^{-5}	5×10^{-5}	5×10^{-5}	5×10^{-5}
Number of epochs	50	50	50	50
Maximum number of gradient steps	$5,000$	$7,500$	$5,000$	$5,000$
β	2.0	2.0	2.0	2.0

Table S7: **Hyperparameters of the iCEM optimizer.** This table reports the optimizer settings used in this work, including the number of samples, horizon length, elite-set size, colored-noise exponent, number of particles, CEM iterations, and the fraction of elites reused. Values are shown separately for each environment and task group.

Hyperparameters	Arm	Fish - (iii) & (iv)	Fish - (v) & (vi)	MSK Leg	SoPrA
Number of samples P	200	200	200	200	200
Horizon H	10	10	100	100	5
Size of elite-set K	20	20	20	20	20
Colored-noise exponent β	0.25	0.25	0.25	0.25	1.0
Number of particles	10	10	10	10	10
CEM-iterations	5	5	5	5	5
Fraction of elites reused ξ	0.3	0.3	0.3	0.3	0.3

Table S8: **Hyperparameters of the SAC algorithm in the model-based RL setting.** This table lists the training hyperparameters used for the SAC agent, including exploration and task horizons, network architecture, ensemble size, batch size, learning rate, and imagined rollout horizon.

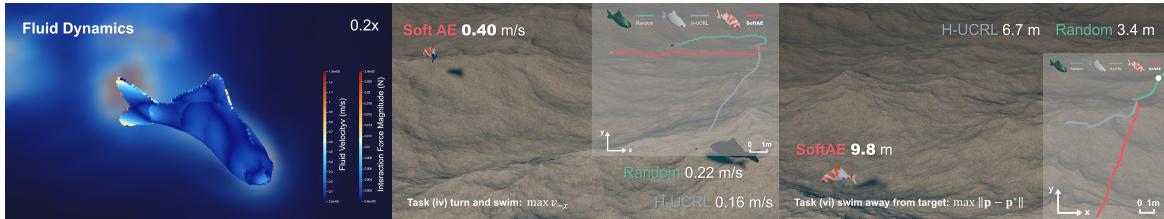| Hyperparameters	Arm
Exploration horizon	200
Downstream task horizon	200
Hidden layers	3
Neurons per layers	1024
Number of ensembles	5
Batch size P	256
Learning rate	3×10^{-4}
Imagined rollout horizon H	5

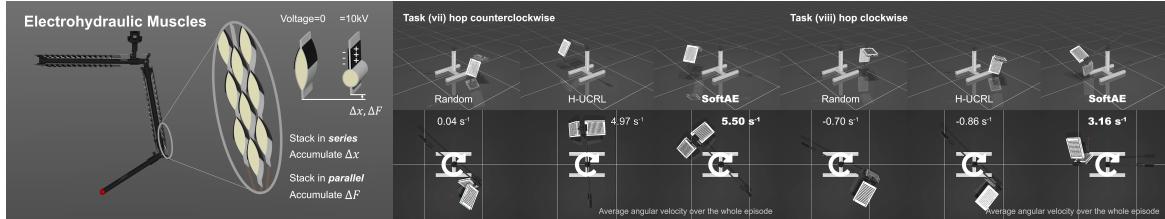...H-UCRL struggles to generalize. ...random exploration is insufficient. Active Exploration (**SoftAE**) learns to explore efficiently enabling success even in difficult tasks.

Caption for Movie S1. Simulated soft continuum arm via Cosserat rods. This video first presents the environment setup, including the state and action space decomposition (see Table S1), before demonstrating the downstream reaching tasks *(i)–(ii)*. Agent performance is compared across three exploration strategies: RANDOM, H-UCRL, and our proposed SOFTAE. The video further shows how downstream task performance improves steadily as more exploration episodes are collected with SOFTAE. To minimize file number, supplementary videos are grouped by soft robotic environments for the initial submission. If needed, future versions can split and provide separate short sequences. Video available at: `https://youtu.be/8hvBvkHiU0g`.

Caption for Movie S2. Articulated fish with deformable skin in fluid simulation. This video follows the same structure as Movie S1, beginning with the environment setup which comprises both the state-action space decomposition (see Table S2), and the particle-based fluid simulation, illustrating fluid dynamics and fluid–structure interaction. It then demonstrates the downstream swimming tasks *(iii)–(vi)*, with performance compared across RANDOM, H-UCRL, and the proposed SOFTAE, as in Movie S1. Video available at: `https://youtu.be/7AykVWWxQq0`.

Caption for Movie S3. **Simulated musculoskeletal leg powered by electrohydraulic muscles.** This video begins by illustrating the electrohydraulic muscle actuation, followed by the system setup, and the state–action space decomposition (see Table S3). It then demonstrates the downstream hopping tasks *(vii)–(viii)*, with performance compared across RANDOM, H-UCRL, and the proposed SOFTAE, as in Movie S1. Video available at: `https://youtu.be/oY4g1fq6lM4`.

Caption for Movie S4. **Real-world pneumatically actuated SoPrA arm.** This video presents the real-world SoPrA environment and task performance and is the animated equivalent of Figure 4. Video available at: `https://youtu.be/JRQ-4fM1yVc`.