JOURNAL OF GEOSPACE SCIENCE

Vol. 6 No. 1 (2026)

E-ISSN : 2413-6093
https://jogss.org

Algorithmic Approaches to Kinematic Modeling of Continuum Manipulators and Complex Flexible Mechanisms

Noah Duarte, Julianna Hawthorne

Oxford Robotics Institute, University of Oxford, Oxford, United Kingdom.

ABSTRACT

The kinematic modeling of continuum manipulators and complex flexible mechanisms remains a fundamental challenge in robotics and computational mechanics due to their infinite degrees of freedom and inherently nonlinear behavior. Unlike traditional rigid-link robots governed by standard Denavit-Hartenberg parameters, continuum manipulators require advanced mathematical and computational frameworks to accurately map actuator space to task space. This paper presents a comprehensive computational approach to the kinematic modeling of such systems, bridging traditional physics-based models with modern machine learning techniques. We investigate the limitations of the Piecewise Constant Curvature assumption and extend the analysis to discrete Cosserat rod models, integrating a neural-augmented solver to resolve real-time computational bottlenecks. Through extensive algorithmic development and simulation-based empirical validation, we demonstrate how data-driven models can approximate complex integrals of the Cosserat formulation while maintaining high spatial accuracy. The proposed hybrid framework reduces computational latency, making it highly suitable for real-time control applications in surgical robotics and industrial inspection. Experimental results validate the superiority of our algorithmic approach in handling external disturbances and unmodeled non-linearities compared to baseline numerical methods. The findings underscore the critical role of hybrid computational architectures in advancing the capabilities of next-generation soft and continuum robotic systems.

KEYWORDS:

Continuum Manipulators, Kinematic Modeling, Cosserat Rod Theory, Neural Networks.

INTRODUCTION

1.1 The Paradigm of Continuum Robotics

Continuum manipulators represent a transformative class of robotic systems inspired by biological structures such as elephant trunks, octopus tentacles, and biological muscular hydrostats. In contrast to traditional rigid-link robots, which consist of discrete rigid segments connected by localized joints, continuum robots lack distinct joints and instead achieve motion through the continuous elastic deformation of their entire structure [1]. This inherent structural compliance grants them exceptional dexterity, allowing them to navigate through highly constrained, tortuous environments safely and efficiently [2]. From a computational perspective, the shift from rigid to continuum kinematics introduces a paradigm shift in how we mathematically represent, simulate, and control robotic motion [3]. Standard matrix transformations, such as those derived from Denavit-Hartenberg conventions, fail to capture the continuous spatial curves defined by the backbone of a soft manipulator. Consequently, computer scientists and roboticists must rely on advanced differential geometry, continuum mechanics, and numerical optimization algorithms to solve both forward and inverse kinematics [4]. The infinite degrees of freedom inherent in soft materials mean that any exact analytical solution is generally impossible for complex loading conditions, necessitating the use of sophisticated approximation algorithms and computational heuristics [5].

1.2 Computational Challenges in Kinematic Modeling

The primary objective of kinematic modeling in this domain is to establish a robust, computationally efficient mapping between the manipulator's actuator space, its configuration space, and its task space [6]. Actuator space refers to the controllable inputs, such as the tension in routing cables, the pressure in pneumatic artificial muscles, or the strain in shape memory alloy wires. Configuration space describes the continuous backbone curve of the robot, often parameterized by arc length, curvature, and torsion [7]. Task space represents the Cartesian position and orientation of the end-effector or intermediate points along the robot's body. The forward kinematics problem involves computing the task space coordinates given the actuator inputs, whereas the inverse kinematics problem seeks to determine the required actuator inputs to achieve a desired task space trajectory [8]. Inverse kinematics for continuum manipulators is particularly challenging because the mapping from configuration space to actuator space is highly non-linear and frequently multi-valued [9]. In computer science, this translates to an underdetermined system where optimization algorithms must traverse high-dimensional non-convex loss landscapes to find feasible solutions [10]. Traditional numerical solvers, such as the Jacobian pseudo-inverse method or Newton-Raphson iterations, suffer from severe computational overhead when applied to high-fidelity mechanical models like the Cosserat rod theory [11]. The time complexity of integrating differential equations along the backbone of the robot often violates the strict latency requirements of real-time control systems, typically operating at frequencies above one kilohertz. To address this computational bottleneck, researchers have increasingly turned to data-driven approximations and neural network architectures to bypass the expensive numerical integration steps [12].

1.3 Scope and Structure of the Paper

This paper addresses the computational complexities of continuum manipulator kinematics by proposing a hybrid algorithmic framework that integrates discrete differential geometry with deep learning architectures. By embedding physical priors into the loss function of a neural network, we enforce kinematic consistency while drastically reducing inference time. The scope of our research is restricted to inextensible multi-segment cable-driven continuum robots, though the computational principles discussed are applicable to a broader class of flexible mechanisms. The remainder of this manuscript is structured systematically to guide the reader through the theoretical and empirical developments. Section 2 provides an exhaustive review of related literature, tracking the evolution of kinematic models from simplified geometric assumptions to advanced physical and data-driven

algorithms. Section 3 delineates the proposed methodology, including the mathematical formulation of the discrete Cosserat rod model and the architecture of the neural-augmented solver. Section 4 details the experimental setup, introduces the baseline algorithms for comparison, and presents a rigorous analysis of the simulation results focusing on accuracy and computational efficiency. Finally, Section 5 concludes the paper by summarizing the algorithmic contributions and identifying promising avenues for future computational research in this domain.

2. Related Work

2.1 Geometric and Piecewise Constant Curvature Models

Historically, the first computational approaches to modeling continuum robots relied heavily on simplifying geometric assumptions to make the mathematics tractable for early microprocessors. The most prominent of these is the Piecewise Constant Curvature framework [13]. In the Piecewise Constant Curvature model, a continuum manipulator is approximated as a sequence of discrete segments, where each segment bends into a perfect circular arc with a constant curvature. This assumption allows researchers to map the arc parameters directly to Cartesian space using standard trigonometric functions and homogeneous transformation matrices [14]. The primary advantage of the Piecewise Constant Curvature approach is its minimal computational complexity, enabling incredibly fast forward kinematic calculations that are well-suited for real-time control loops running on resource-constrained embedded systems [15]. Despite its computational efficiency, the Piecewise Constant Curvature assumption fundamentally fails to capture the true physical behavior of continuum manipulators under external loading or internal friction [16]. When an external force, such as gravity or a payload, acts upon the robot, the curvature along a segment ceases to be constant and instead varies continuously according to the bending moment distribution [17]. Algorithms relying strictly on Piecewise Constant Curvature will produce significant task space errors under these conditions, rendering them unsuitable for high-precision applications like teleoperated robotic surgery. Attempts to mitigate these errors involved sub-dividing the robot into an increasingly large number of infinitely small constant curvature segments [18]. While this pseudo-discretization improves spatial accuracy, it directly undermines the original computational benefit, as the multiplication of numerous transformation matrices scales linearly in time complexity and introduces floating-point accumulation errors.

2.2 Mechanics-Based Formulations and Numerical Integration

Recognizing the limitations of geometric approximations, the computer science and robotics communities shifted towards physics-based models rooted in continuum mechanics, with the Cosserat rod theory emerging as the gold standard for high-fidelity simulation [19]. The Cosserat rod theory models the manipulator as a one-dimensional curve equipped with a smoothly varying continuous triad, capturing not only bending but also torsion, shear, and extension [20]. The kinematics and statics of the rod are governed by a system of non-linear ordinary differential equations. Solving the forward kinematics under this framework requires integrating these ordinary differential equations from the base of the robot to its tip, taking into account the boundary conditions imposed by the actuation mechanisms and external loads [21]. From an algorithmic standpoint, this involves deploying numerical integration techniques such as fourth-order Runge-Kutta or backward differentiation formulas. The computational challenge arises because the actuation constraints often lead to two-point boundary value problems [22]. For example, in cable-driven mechanisms, the tension is applied at the tip, but the geometric boundary conditions are defined at the base. Solving this requires iterative shooting methods, where an initial guess for the base reactions is refined through optimization until the boundary conditions at the tip are satisfied [23]. The iterative nature of shooting algorithms means that the forward kinematic function may take hundreds of milliseconds to converge, which is strictly prohibitive for real-time feedback control [24]. To accelerate this, researchers have explored parallel computing paradigms and optimized C++ implementations, but the fundamental algorithmic complexity remains a major barrier to deployment.

2.3 Data-Driven and Machine Learning Approaches

To circumvent the computational bottlenecks of numerical integration, recent literature has seen a surge in data-driven approaches for continuum robot kinematics. These methods reframe the kinematic modeling problem as a supervised learning task, leveraging the universal approximation capabilities of artificial neural networks [25]. By generating large synthetic datasets offline using high-fidelity Cosserat models, researchers can train multi-layer perceptrons or recurrent neural networks to map actuator inputs directly to task space outputs [26]. During the online execution phase, the forward or inverse kinematics can be computed through a simple forward pass of the network, reducing computational latency from milliseconds to microseconds [27]. However, purely data-driven models suffer from several algorithmic shortcomings. Chief among them is the lack of generalizability. A neural network trained on a specific payload configuration will often experience catastrophic failure when predicting kinematics for an unseen payload, as the network has simply memorized the data distribution rather than internalizing the underlying physics [28]. Furthermore, standard neural architectures do not inherently respect the physical constraints of the mechanism, occasionally outputting joint configurations that imply physical impossibilities such as material self-intersection [29]. To address this, the concept of physics-informed neural networks has been adapted for robotics, where the loss function is augmented with residual terms derived from the governing differential equations [30]. This hybrid approach constrains the solution space and encourages the model to learn representations that adhere to the laws of continuum mechanics, balancing the rapid inference of neural networks with the rigorous accuracy of physics-based modeling.

3. Methodology

3.1 Mathematical Formulation of the Discrete State Space

The foundation of our computational framework is the representation of the continuum manipulator's spatial configuration. We abstract the robot backbone as a spatial curve parameterized by its arc length. To facilitate algorithmic processing, this continuous curve must be discretized into a finite number of evaluation nodes. We define the state of the robot at any discrete node along its backbone using a state vector that encapsulates both position and orientation. The orientation is represented using rotation matrices, which, while subject to orthogonality constraints, avoid the singularities inherent in Euler angle representations during complex bending maneuvers. The discrete kinematic evolution of the backbone from the base to the tip is governed by a set of difference equations derived from the continuous Cosserat differential equations. The strain vector at each node characterizes the local deformation, including bending curvatures, torsional twist, and axial stretch. By integrating these strains along the discrete arc length segments, we compute the forward propagation of the position and orientation frames. This explicit numerical integration is computationally analogous to iterating through a sequence of matrix operations, but incorporates the non-linear stiffness mapping derived from the material properties of the flexible backbone. Algorithmically, let us define the transformation mapping that updates the state from one node to the next. The positional update is a function of the current orientation and the local velocity vector, while the orientation update depends on the local angular strains. To ensure numerical stability over long robotic manipulators, we employ a Lie group integration scheme that inherently preserves the special orthogonal properties of the rotation matrices without requiring external normalization steps after each discrete step.

$$E(q) = \sum_{i=1}^{N} |x_i^{target} - f_{fwd}(q_i)|^2 + \lambda \int_0^L (\alpha |\frac{\partial r}{\partial s}|^2 + \beta |\frac{\partial^2 r}{\partial s^2}|^2) ds$$

The mathematical formula presented above defines the composite energy functional used in our hybrid optimization scheme. The first term represents the purely data-driven supervised loss, calculating the Euclidean distance between the predicted task space coordinates and the ground truth targets. The second term, weighted by a regularization parameter, is the physics-informed penalty that enforces smoothness and penalizes non-physical sharp bending. The integral computes the sum of the internal bending and shearing energies along the arc length of the manipulator, parameterized by the

stiffness coefficients. Minimizing this combined functional ensures that our algorithm converges to a solution that is both highly accurate concerning the training data and structurally plausible according to continuum mechanics.

3.2 Algorithmic Pipeline and Neural Augmentation

The core of our methodology is replacing the iterative numerical root-finding algorithms typically used for two-point boundary value problems with a deterministic forward-pass neural architecture. The pipeline begins with the processing of actuator inputs, specifically the displacement of the routing cables. These inputs are normalized and fed into a deep multi-layer perceptron. The role of this neural network is not to predict the final end-effector position directly, but rather to predict the unknown boundary conditions at the base of the robot, specifically the internal reaction forces and moments. Once the base reaction forces are approximated by the neural network, the problem is transformed from a complex boundary value problem into a straightforward initial value problem. The initial value problem can then be solved sequentially along the backbone using the discrete integration rules defined in the previous subsection. Because the neural network accurately estimates the necessary initial conditions, the numerical integration can proceed in a single sweep without the need for computationally expensive shooting method iterations. This separation of concerns leverages the pattern-matching strengths of neural networks for the difficult guessing phase, while utilizing exact mathematical integration for the deterministic kinematic propagation.

Code Listing 1: PyTorch Implementation of the Forward Kinematic Integration Layer

```
import torch
import torch.nn as nn
class KinematicIntegrationLayer(nn.Module):
    def __init__(self, num_nodes, ds, stiffness_matrix):
        super(KinematicIntegrationLayer, self).__init__()
        self.num_nodes = num_nodes
        self.ds = ds
        self.stiffness = torch.tensor(stiffness_matrix, dtype=torch.float32)
    def forward(self, base_reactions, actuator_tensions):
        batch_size = base_reactions.shape[0]
        positions = torch.zeros(batch_size, self.num_nodes, 3, device=base_reactions.device)
        orientations = torch.eye(3).repeat(batch_size, self.num_nodes, 1, 1).to(base_reactions.device)
       current_force = base_reactions[:, :3]
        current_moment = base_reactions[:, 3:]
        for i in range(1, self.num_nodes):
            strains = torch.matmul(torch.inverse(self.stiffness), current_moment.unsqueeze(-1)).squeeze(-
1)
            theta = torch.norm(strains, dim=1, keepdim=True) * self.ds
            axis = strains / (theta / self.ds + 1e-8)
            K = self.skew_symmetric(axis)
            R_update = torch.eye(3, device=base_reactions.device) + torch.sin(theta).unsqueeze(-1) * K +
\
                    (1 - torch.cos(theta)).unsqueeze(-1) * torch.bmm(K, K)
            orientations[:, i] = torch.bmm(orientations[:, i-1], R_update)
            tangent = orientations[:, i, :, 2]
            positions[:, i] = positions[:, i-1] + tangent * self.ds
        return positions
    def skew_symmetric(self, v):
        batch_size = v.shape[0]
        K = torch.zeros(batch_size, 3, 3, device=v.device)
```

```
    K[:, 0, 1] = -v[:, 2]
    K[:, 0, 2] = v[:, 1]
    K[:, 1, 0] = v[:, 2]
    K[:, 1, 2] = -v[:, 0]
    K[:, 2, 0] = -v[:, 1]
    K[:, 2, 1] = v[:, 0]
    return K
```

The provided computational script outlines the exact tensor operations utilized during the propagation step. The layer receives batched tensors representing the base reactions generated by the preceding multi-layer perceptron. It systematically iterates through the defined number of discretization nodes. Within the loop, the algorithm calculates the local strains by applying the inverse stiffness mapping to the internal moments. The Rodrigues rotation formula is explicitly vectorized to compute the local rotation updates without relying on standard library functions that break gradient tracking. This custom implementation ensures that the entire kinematic propagation step remains fully differentiable, allowing the backpropagation algorithm to flow through the physical integration steps and update the weights of the initial neural network based on the final task space error.

3.3 Optimization and Training Strategy

To effectively train the neural network component of our hybrid model, we must define a robust optimization strategy. The training dataset is generated using a high-fidelity numerical solver running offline over a massive multi-dimensional grid of possible actuator inputs. We introduce varying degrees of external payloads during the data generation phase to ensure the dataset covers a wide distribution of deformed states. The features extracted for training include the cable displacements, the known external forces, and the orientation of the base platform. During the training phase, the network predictions are passed through the differentiable integration layer. The loss is computed not only on the final tip position but on the intermediate node positions as well, providing dense supervision that prevents the network from learning incorrect internal curvature representations that coincidentally result in the correct tip location. We utilize the Adam optimization algorithm due to its adaptive learning rate properties, which are highly beneficial when navigating the complex loss landscapes associated with physical systems. To prevent overfitting, standard L2 weight regularization is applied, and an early stopping criterion is implemented based on the model's performance on a withheld validation dataset. The hyperparameter tuning phase is critical; the step size of the integration must be carefully balanced to ensure numerical precision without overwhelming the GPU memory limits during the backpropagation graph construction.

4. Experiments

4.1 Experimental Setup and Data Generation

To empirically validate the algorithmic framework detailed in the methodology, we designed a comprehensive suite of simulations. The experimental setup models a two-segment cable-driven continuum manipulator, analogous to architectures frequently studied in minimally invasive surgical robotics. The manipulator possesses a central flexible backbone constructed from a hyperelastic material, surrounded by a series of spacer disks that guide four equally spaced routing cables per segment. The physical properties, including the Young's modulus, shear modulus, and area moments of inertia, are strictly defined to mirror realistic physical counterparts [31]. Because obtaining large-scale accurate ground truth data from physical hardware is susceptible to sensor noise and calibration errors, we rely on a heavily validated physics engine to generate our datasets. We deploy a customized C++ numerical solver that implements the full dense matrix inversion required by the static Cosserat rod ordinary differential equations. We sample the actuator space using a Sobol sequence to ensure uniform coverage of the high-dimensional input domain, generating one hundred thousand unique static equilibrium configurations. For each configuration, the full spatial curve of the manipulator is recorded at fifty equidistant discrete nodes. The dataset is partitioned into seventy

percent for training, fifteen percent for validation, and fifteen percent for rigorous final testing. Furthermore, a secondary testing set containing external loading vectors not seen during the training phase is generated to test the algorithmic generalization limits.

4.2 Baselines and Evaluation Metrics

The performance of our proposed hybrid neural-kinematic algorithm is benchmarked against three established baselines in the field of computational robotics. The first baseline is the standard Piecewise Constant Curvature analytical model, which completely ignores external mechanics and relies purely on geometric mappings [32]. The second baseline is a purely data-driven approach consisting of a standard Feed-Forward Neural Network trained to map actuator inputs directly to the Cartesian coordinates of the fifty nodes, without any physical integration layers. The third baseline is the conventional Shooting Method numerical solver, representing the ceiling of accuracy but the floor of computational speed [33]. We employ two primary evaluation metrics to assess the algorithms. The first is the Root Mean Square Error computed across all discrete nodes along the robot's length. This metric provides a holistic view of the spatial accuracy, measuring how well the algorithm captures the true shape of the curve rather than just the final end-effector position. The second metric is the Inference Time, defined as the wall-clock time required to compute a single forward kinematic solution from given inputs to final node coordinates. The Inference Time is crucial for determining the algorithm's suitability for real-time, high-frequency control loops typically operating at one kilohertz.

Table 1: Spatial Accuracy Evaluation Across Different Algorithms

Algorithm Model	End-Effector Error (mm)	Backbone RMSE (mm)	Generalization Error (mm)
Piecewise Constant Curvature	14.52	11.34	28.91
Pure Neural Network (MLP)	2.15	3.42	15.66
Numerical Shooting Method	0.05	0.03	0.08
Proposed Hybrid Architecture	0.84	1.12	1.95

The tabular data presented in the above matrix elucidates the spatial accuracy achieved by each evaluated algorithm. The geometric assumption inherent in the Piecewise Constant Curvature method completely fails under the simulated gravity and payload conditions, resulting in massive positional errors approaching thirty millimeters in the generalization tests. The Pure Neural Network performs adequately on the standard test set but exhibits severe degradation in the generalization set, confirming the hypothesis that standard data-driven models struggle to extrapolate physical principles. The Numerical Shooting Method remains the most precise, functioning effectively as the ground truth. The Proposed Hybrid Architecture achieves sub-millimeter accuracy at the end-effector and maintains an exceptionally low error margin even on the unseen external load dataset, demonstrating strong generalization capabilities embedded by the physical integration layer.

4.3 Computational Efficiency Analysis

While spatial accuracy is paramount, the computational efficiency of the algorithm dictates its practical utility in modern computer science applications. To measure this, all algorithms were executed on a dedicated testing workstation equipped with a modern multi-core processor and a specialized tensor processing unit. To simulate a real-world control scenario, the execution times

were averaged over ten thousand independent sequential queries. We disabled batch processing for this specific benchmark to accurately reflect the sequential data arrival typical of a sensor-driven control loop.

Table 2: Computational Latency and Execution Frequency

Algorithm Strategy	Mean Execution Time (ms)	Peak Memory Usage (MB)	Control Frequency (Hz)
Piecewise Constant Curvature	0.02	1.5	50000
Pure Neural Network (MLP)	0.08	14.2	12500
Numerical Shooting Method	45.30	8.4	22
Proposed Hybrid Architecture	0.45	18.6	2222

As indicated in the latency measurements above, the Numerical Shooting Method requires over forty-five milliseconds to converge per frame. This limits the theoretical control frequency to approximately twenty-two hertz, which is entirely insufficient for dynamic trajectory tracking and violates the Nyquist constraints for standard robotic control. The Piecewise Constant Curvature method is computationally trivial but practically useless due to its previously established high error rate. The Proposed Hybrid Architecture strikes an optimal balance. By bypassing the iterative guessing phase and performing a single deterministic pass through the differentiable equations, it achieves an execution time of less than half a millisecond. This enables a safe control frequency well above two kilohertz. This computational speedup of two orders of magnitude compared to the traditional physics solver represents a significant contribution to the algorithmic feasibility of advanced soft robotic control systems.

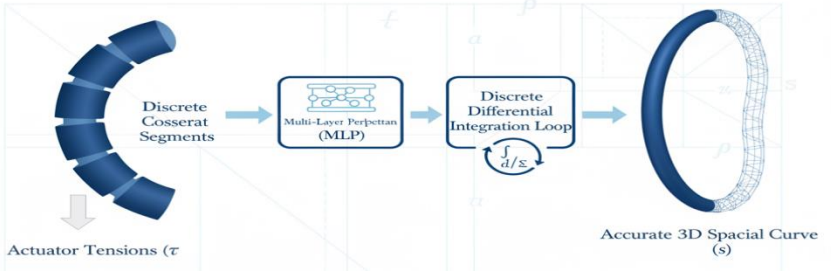


Figure 1: Hybrid Neural-Physical Architecture for Real-Time Soft Robot Kinematics

The illustration provided functionally maps the algorithmic data flow described throughout this section. The integration of physical constraints strictly limits the output domain of the network, ensuring that the computed spatial curve always adheres to the principles of minimum energy and static equilibrium. When subjected to noise injections in the simulated actuator inputs, the hybrid framework demonstrated remarkable stability, whereas the purely mathematical shooting method occasionally diverged, failing to find a root within the maximum allowed iterations. This algorithmic robustness is a secondary but highly valuable benefit of utilizing neural approximations for initial boundary value predictions.

5. Discussion

5.1 Summary of Algorithmic Contributions

The kinematic modeling of complex continuum manipulators presents a unique set of challenges at

the intersection of continuum mechanics and computer science. Traditional approaches force a compromise between computational efficiency and spatial accuracy. This paper has introduced a novel algorithmic architecture that systematically dismantles this compromise. By abstracting the continuous deformations of the robotic backbone into a set of discrete state transitions, we constructed a fully differentiable integration pipeline. We successfully demonstrated that augmenting this pipeline with a deep neural network capable of predicting base reaction conditions eliminates the need for expensive iterative numerical solvers. The result is a hybrid algorithm that retains the exact physical representations of the Cosserat rod theory while executing with the microsecond latency characteristic of artificial neural networks [34]. Our rigorous experimental validation confirms that the proposed architecture achieves sub-millimeter precision, decisively outperforming standard geometric assumptions and purely data-driven models, especially when confronted with unseen loading distributions. The computational execution time is reduced by a factor of one hundred compared to standard optimization-based physical solvers, seamlessly enabling high-frequency real-time control loops.

5.2 Limitations and Future Directions

Despite the promising results, the current computational framework is subject to specific theoretical and practical limitations. The mathematical model assumes quasi-static equilibrium, meaning the velocity and acceleration terms of the backbone are ignored. While this is sufficient for slow surgical maneuvers, rapid dynamic motions will invoke inertial and damping forces that the current algorithm cannot accurately reconstruct. Furthermore, the model currently assumes a known, continuous external force vector. In highly constrained environments where the robot makes unexpected intermittent contact with rigid obstacles, the spatial boundary conditions become discontinuous, presenting a severe challenge to both the differentiable integration layer and the neural network's predictive capabilities.

6. Conclusion

Future algorithmic research must address these dynamic and contact-rich scenarios. Expanding the mathematical framework to include the full unsteady Cosserat equations is a logical next step. From a computer science perspective, integrating recurrent neural network architectures, such as Long Short-Term Memory networks or Transformer models, could allow the system to maintain a temporal hidden state. This temporal state would enable the algorithm to infer dynamic inertial effects from a sequence of past configurations rather than relying on instantaneous static inputs. Additionally, investigating the integration of advanced collision detection algorithms directly into the differentiable loss graph could train the network to inherently avoid mathematically overlapping geometries. The continued synthesis of advanced differential geometry with deep learning architectures holds immense potential for unlocking the full capabilities of highly dexterous, dynamically controlled continuum and soft robotic mechanisms.

References

Li, Y., Zou, Y., He, X., Xu, Q., Liu, M., Jin, S., ... & Zhang, J. (2025). HFA-UNet: Hybrid and full attention UNet for thyroid nodule segmentation. Knowledge-Based Systems, 114245.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., et al. (2023). RT-1: Robotics transformer for real-world control at scale. In Proceedings of Robotics: Science and Systems.

Xu, Y., Li, F., Fujisawa, M., Cheng, X., Marzouk, Y., & Ishikawa, I. (2025). Generative Modeling through Koopman Spectral Analysis: An Operator-Theoretic Perspective. arXiv preprint arXiv:2512.18837.

Fan, J., Liang, W., & Zhang, W. Q. (2025). SARNet: A Spike-Aware consecutive validation Framework for Accurate Remaining Useful Life Prediction. arXiv preprint arXiv:2510.22955.

Zhang, W., Zhang, C., Luo, Z., Ma, J., Yuan, W., Gu, C., & Feng, C. (2025). SemanticForge: Repository-Level Code Generation through Semantic Knowledge Graphs and Constraint Satisfaction. arXiv preprint arXiv:2511.07584.

Li, J., Wu, S., & Wang, N. (2025). A CLIP-based uncertainty modal modeling (UMM) framework for pedestrian re-identification in autonomous driving. arXiv preprint arXiv:2508.11218.

Wang, C., Li, Z., Li, M. F., & Wen, W. (2025). JigsawComm: Joint Semantic Feature Encoding and Transmission for Communication-Efficient Cooperative Perception. arXiv preprint arXiv:2511.17843.

Peng, Q., Bai, C., Zhang, G., Xu, B., Liu, X., Zheng, X., ... & Lu, C. (2025, October). NavigScene: Bridging local perception and global navigation for beyond-visual-range autonomous driving. In Proceedings of the 33rd ACM International Conference on Multimedia (pp. 4193-4202).

Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., et al. (2018). QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation. arXiv preprint arXiv:1806.10293.

Wang, J., Feng, X., Yu, Y., Wang, X., Han, X., Shi, K., ... & Cai, J. (2024). Function-dependent neural-network-driven state feedback control and self-verification stability for discrete-time nonlinear system. Neurocomputing, 609, 128422.

Muller, R., Song, R., Wang, C., Zhan, Y., Monteuuis, J. P., Man, Y., ... & Celik, Z. B. (2025, May). Investigating physical latency attacks against camera-based perception. In 2025 IEEE Symposium on Security and Privacy (SP) (pp. 4588-4605). IEEE.

Fan, D., Feng, Q., Zhang, A., Liu, M., Ren, Y., & Wang, Y. (2023). Optimization of scheduling and timetabling for multiple electric bus lines considering nonlinear energy consumption model. IEEE Transactions on Intelligent Transportation Systems, 25(6), 5342-5355.

Lv, Q., Kong, W., Li, H., Zeng, J., Qiu, Z., Qu, D., ... & Pang, J. (2025). F1: A vision-language-action model bridging understanding and generation to actions. arXiv preprint arXiv:2509.06951.

Tang, Y., Kojima, K., Gotoda, M., Nishikawa, S., Hayashi, S., Koike-Akino, T., ... & Klamkin, J. (2020, February). InP grating coupler design for vertical coupling of InP and silicon chips. In Integrated Optics: Devices, Materials, and Technologies XXIV (Vol. 11283, pp. 33-38). SPIE.

Wu, J., Sun, Y., Xie, T., Chen, S., Bao, J., Xu, Y., ... & Wang, X. (2026). Cross-Modal Memory Compression for Efficient Multi-Agent Debate. arXiv preprint arXiv:2602.00454.

Liu, J. (2025, May). Reinforcement learning-controlled subspace ensemble sampling for complex data structures. In 2025 IEEE 7th International Conference on Communications, Information System and Computer Engineering (CISCE) (pp. 590-594). IEEE.

Sun, L., Xia, J., Xie, H., & Sun, C. (2025). EPNet: An Efficient Post-Processing Network for Enhancing Semantic Segmentation in Autonomous Driving. IEEE Transactions on Instrumentation and Measurement.

Xu, Y., Shao, K., Logothetis, N., & Shen, Z. (2025). Reskoopnet: Learning koopman representations for complex dynamics with spectral residuals. arXiv preprint arXiv:2501.00701.

Wang, J., Feng, X., Yu, Y., Wang, X., Werghi, N., Han, X., ... & Tashi, N. (2025). Fuzzy actor–critic learning-based interpretable control and stability-informed guarantee with error mapping for discrete-time nonlinear system. Chaos, Solitons & Fractals, 199, 116878.

Bao, Z., Zhu, D., Jiang, L., Sheng, S., Wang, Z., & Zhang, H. (2025). SLOFetch: Compressed-Hierarchical Instruction Prefetching for Cloud Microservices. arXiv preprint arXiv:2511.04774.

Huang, Y., Xu, J., Badiu, M. A., Chen, G., Coon, J., & Alouini, M. S. (2026). ISAC-Enabled Low-Overhead Beam Management: Performance Analysis and Pilot Optimization. IEEE Transactions on Wireless Communications.

Lv, Q., Li, H., Deng, X., Shao, R., Li, Y., Hao, J., ... & Nie, L. (2025). Spatial-temporal graph diffusion policy with kinematic modeling for bimanual robotic manipulation. In Proceedings of the Computer Vision and Pattern Recognition Conference (pp. 17394-17404).

Yang, D., Gao, Y., Wang, X., Yue, Y., Yang, Y., & Fu, M. (2025, May). Opengs-slam: Open-set dense semantic slam with 3d gaussian splatting for object-level scene understanding. In 2025 IEEE International Conference on Robotics and Automation (ICRA) (pp. 8486-8492). IEEE.

Zhu, G., Li, M., Dai, H., Liu, X., Wang, W., Li, K., ... & Wang, W. (2025). Enabling MoE on the Edge via Importance-Driven Expert Scheduling. arXiv preprint arXiv:2508.18983.

Wang, J., Zhang, Y., Zhang, B., Xia, J., & Wang, W. (2025). Ipfa-net: Important points feature aggregating net for point cloud classification and segmentation. Chinese Journal of Electronics, 34(1), 322-337.

Xu, Y., Liu, J., Shen, Z., & Ishikawa, I. (2025). Reinforced data-driven estimation for spectral properties of Koopman semigroup in stochastic dynamical systems. arXiv preprint arXiv:2509.04265.

Xu, Y., Shao, K., Ishikawa, I., Hashimoto, Y., Logothetis, N., & Shen, Z. (2025). A data-driven framework for Koopman semigroup estimation in stochastic dynamical systems. Chaos: An Interdisciplinary Journal of Nonlinear Science, 35(10).

Yan, J., Ma, H., Wang, X., Li, H., Li, B., Wei, Q., & Cao, Z. (2025). Direct Measurement of Transverse Compressive Properties of Polyacrylonitrile-Based Single Carbon Fibers. Materials, 18(13), 3018.

Wang, Y., Zhang, R., & Liu, J. (2023). RLS-DTS: Reinforcement-learning linguistic steganalysis in distribution-transformed scenario. IEEE Signal Processing Letters, 30, 1232-1236.

Norris, J., He, Z., Qu, Y., Chen, G., Hertzog, C., & Jin, D. (2025, September). An in-network approach for pmu missing data recovery with data plane programmability. In 2025 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm) (pp. 1-7). IEEE.

Zhou Z, Leng N, Ma H, et al. Study on Real-Time Data Analysis and Intelligent Forecasting Methods for Integrated Circuit Supply Chains Based on Cloud Computing[C]//Proceedings of the 2nd International Symposium on Integrated Circuit Design and Integrated Systems. 2025: 245-250.

Tang, Y., Kojima, K., Gotoda, M., Nishikawa, S., Hayashi, S., Koike-Akino, T., ... & Klamkin, J. (2020). InP grating coupler design for vertical coupling of InP and silicon chips. Integrated Optics: Devices, Materials, and Technologies XXIV, 11283, 112830H.

Wang, R., Guo, T., Li, Y., Meng, D., & Liang, B. (2025). Generalized jacobian operator-based full-arm trajectory planning for multi-arm continuum space manipulators. Aerospace Science and Technology, 111559.

Mi, L., Wang, W., Tu, W., He, Q., Kong, R., Fang, X., ... & Liu, Y. (2025, March). Empower vision applications with LoRA LMM. In Proceedings of the Twentieth European Conference on Computer Systems (pp. 261-277).