Robust Shape-Free Manipulation through a Graph-Based Reinforcement Learning Approach for Deformable Objects

Mahdi Bonyani [1], Maryam Soleymani[1], and Chao Wang[2]

***Abstract*— Robotic manipulation of deformable objects remains a challenging problem due to high-dimensional geometry, dynamic topology, and limited actuation. We propose a novel reinforcement learning framework that models the manipulation scene as a heterogeneous graph and leverages SE(3)-equivariant message passing to enable shape-free, generalizable control. Our method, GRPD, combines structured graph design, spatially-aware policy networks, and bounded policy optimization for stable learning. Across diverse tasks such as rope shaping and cloth hanging, GRPD outperforms baseline policies in terms of sample efficiency, robustness to noise, and generalization to unseen objects.**

I. INTRODUCTION

Manipulating deformable objects represents an important frontier in robotics research, with wide-ranging applications in manufacturing, domestic services, and healthcare settings [1], [2]. Unlike rigid objects, deformable objects pose unique challenges for robotic manipulation. They are characterized by high-dimensional state spaces and complex, nonlinear dynamics that make state estimation difficult and forward prediction computationally expensive [2]. The pose of a deformable object is insufficient as a state representation for manipulation tasks, as the object's shape changes in response to manipulation actions [3], [4], [5]. This fundamental difference from rigid object manipulation necessitates specialized approaches to modeling and control. To solve these challenges, some methods focus on learning models directly in image space or latent space without incorporating physics priors [6], [7]. These approaches often suffer from low data efficiency and limited generalization capabilities. Other researchers have developed explicit state-space representations that incorporate physics priors about deformable object behavior, such as mass-spring systems reflected in network structures [2]. So, a better approach to deformable object manipulation requires an object model that integrates both shape representation and prediction capabilities. This integration is essential for enabling robots to perform both low-level tasks like pick-and-place operations and high-level manipulation tasks that require planning and hierarchical reasoning [5].

Graph-based representations have become increasingly popular for modeling deformable objects due to their ability to capture complex shapes, dynamics, and interactions in a computationally efficient manner [8]. These representations typically approximate deformable objects as a sparse set of interacting keypoints or particles, which are connected through graph structures that reflect the underlying physical properties and relationships [9]. Graph-based representations also enable the incorporation of physics principles into the modeling of deformable objects. The particle-based nature of these representations offers significant advantages in capturing object dynamics by leveraging the inductive biases of particle systems [10]. Frameworks like AdaptiGraph employ graph neural networks to predict particle motion in a unified physical property-conditioned model capable of simulating diverse materials with varying physical properties [11]. The key contributions of our work include:

- We propose a novel graph-based reinforcement learning framework for deformable object manipulation that integrates $SE(3)$-equivariant [12], [13] message passing with explicit modeling of actuator-object heterogeneity.
- We introduce a symmetry-aware training pipeline using modified bounded optimization and theoretically establish equivariance guarantees and universal approximation for shape-free manipulation policies.

II. METHODOLOGY

This section outlines our proposed method, Graph-Reinforced Policy for Deformables (GRPD), a graph-based reinforcement learning framework that facilitates robust manipulation of deformable objects with arbitrary geometries. Our approach hinges on three central components: (i) a heterogeneous graph representation of the manipulation scene, (ii) an SE(3)-equivariant [12], [13] message-passing policy network, and (iii) a bounded policy optimization scheme that stabilizes learning in high-dimensional state-action spaces.

The manipulation environment is encoded as a dual-node graph, separating actuators and deformable elements, with directed edges capturing both intra- and inter-group interactions. Right – Our GRPD policy processes this structure using symmetry-preserving message passing and aggregates action signals at actuator nodes.

A. Problem Formulation

We formulate the manipulation task as a Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ the action space, $\mathcal{P}$ the transition probability function, $r : \mathcal{S} \times \mathcal{A} \to R$ the reward function, and $\gamma \in [0, 1)$ the discount factor.

At each timestep t, the environment is encoded as a heterogeneous graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ with node set $\mathcal{V}_t =$

[1] are Ph.D. Student, Bert S. Turner Department of Construction Management, Louisiana State University, USA mbonya1@lsu.edu msoley1@lsu.edu

[2]Associate Professor and Graduate Program Advisor, Bert S. Turner Department of Construction Management, Louisiana State University, USA chaowang@lsu.edu

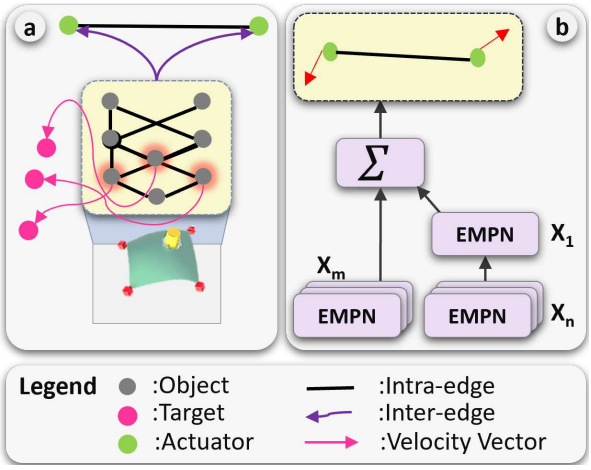

Fig. 1. An overview of graph-structured representation of deformable manipulation. The subfigure (a) shows the manipulation environment is encoded as a dual-node graph, separating actuators and deformable elements, with directed edges capturing both intra- and inter-group interactions. The subfigure (b) shows our GRPD policy processes this structure using symmetry-preserving message passing and aggregates action signals at actuator nodes.

$\mathcal{V}_{\text{obj}} \cup \mathcal{V}_{\text{act}}$, corresponding to deformable object points and actuators respectively. Each node $v \in \mathcal{V}_t$ is associated with a feature vector h_v^t, comprising positional coordinates $p_v^t \in R^3$, velocity $v_v^t \in R^3$, and task-specific features such as distance to target and node type embeddings.

Our goal is to learn a policy $\pi_\theta(a_t|\mathcal{G}_t)$ that selects actions $a_t \in \mathcal{A}$ to maximize the expected return:

$$E_{\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right] \tag{1}$$

under the constraint that π_θ is equivariant with respect to rigid body transformations in SE(3).

B. Heterogeneous Graph Representation

The graph $\mathcal{G}_t$ contains directed edges of different types:

- Intra-object edges $\mathcal{E}_{\text{obj-obj}}$ to encode local connectivity within the deformable structure.
- Intra-actuator edges $\mathcal{E}_{\text{act-act}}$ for modeling interactions between multiple end-effectors.
- Inter-agent-object edges $\mathcal{E}_{\text{act-obj}}$, fully connecting each actuator to all object nodes, ensuring global awareness.

Let h_v^t denote the feature vector at node v at time t. We define edge-wise message functions ψ_e and update functions ϕ_v such that the message-passing update rule is:

$$m_v^{(k)} = \sum_{u\in\mathcal{N}(v)} \psi_e^{(k)}(h_v^{(k)}, h_u^{(k)}, e_{uv}) \tag{2}$$

$$h_v^{(k+1)} = \phi_v^{(k)}(h_v^{(k)}, m_v^{(k)}) \tag{3}$$

where e_{uv} is the edge type, and k indexes the propagation layer.

C. Equivariant Policy Architecture

To respect geometric symmetries, we ensure the policy is SE(3)-equivariant. This means that applying a rigid transformation $g \in \text{SE}(3)$ to the scene transforms the policy output accordingly:

$$\pi_\theta(g \cdot \mathcal{G}) = g \cdot \pi_\theta(\mathcal{G}) \tag{4}$$

To enforce this, we modified an Equivariant Message Passing Network (EMPN) [14], [15], [16] in which each message function ψ_e is constrained to be equivariant:

$$\psi_e(g \cdot h_v, g \cdot h_u, e_{uv}) = g \cdot \psi_e(h_v, h_u, e_{uv}) \tag{5}$$

This is implemented by using relative positions $(p_u - p_v)$ and orientation-aligned convolution kernels κ:

$$\psi_e(h_v, h_u, e_{uv}) = \kappa(p_u - p_v) \cdot h_u \tag{6}$$

The final actuator features h_a are decoded into actions via:

$$a = \pi_\theta(\mathcal{G}_t) = \text{MLP}(h_a) \tag{7}$$

D. Value Function Estimation

The value function $V_\psi(\mathcal{G}_t)$ is constructed using a permutation-invariant DeepSets [17] formulation:

$$V_\psi(\mathcal{G}_t) = \text{MLP}_{\text{out}}\left(\sum_{v\in\mathcal{V}_t}\text{MLP}_{\text{in}}(h_v^t)\right) \tag{8}$$

This allows flexibility in varying graph sizes and node types while maintaining expressiveness.

E. Bounded Policy Optimization

Instead of standard PPO, we adopt Bounded Policy Optimization (BPP) to improve training stability in complex high-dimensional environments. We solve the following constrained optimization:

$$\theta_{k+1} = \arg\max_\theta E_{(s,a)\sim\pi_{\theta_k}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}A_{\pi_{\theta_k}}(s,a)\right] \tag{9}$$

$$\text{s.t. } D_{\text{KL}}(\pi_\theta||\pi_{\theta_k}) \le \delta \tag{10}$$

where $A_{\pi_{\theta_k}}$ is the advantage function, and δ is a divergence bound. BPP performs a differentiable projection step to maintain this bound on both mean and variance of the Gaussian policy output.

F. Theoretical Insight: Information Propagation

We model actuators as virtual nodes with full connectivity to object nodes. Denoting the Jacobian of actuator output w.r.t. object input as $J = \partial a/\partial h^{\text{obj}}$, our design ensures:

$$J_{vu} \neq 0 \quad \forall v \in \mathcal{V}_{\text{act}}, \forall u \in \mathcal{V}_{\text{obj}} \tag{11}$$

within a single message-passing layer. This guarantees that actuator outputs are sensitive to the global state of the deformable object, unlike in traditional GNNs where over-squashing may hinder long-range dependency propagation.

III. Experiments and Results

To evaluate the effectiveness of our proposed GRPD framework, we design a comprehensive of robotic manipulation tasks involving deformable objects of varying topologies and resolutions. Our experimental protocol focuses on answering key research question: Does explicit modeling of heterogeneity and geometric equivariance improve manipulation success?

A. Experimental Setup

All experiments are conducted in NVIDIA IsaacGym/IsaacLab, a GPU-accelerated simulator for physics-based reinforcement learning. We simulate deformable objects such as cloth sheets and ropes with varying levels of discretization (from 20 to 1200 nodes), manipulated by 2–4 end-effectors. Each manipulation scenario is modeled as a heterogeneous graph with actuator and object nodes, with spatial relationships captured through relative positions and velocities.

We benchmark GRPD against three policy architectures:

- **Transformer Policy**: A non-equivariant attention-based model treating all nodes homogeneously [18].
- **Equivariant MPN**: A homogeneous SE(3)-equivariant message-passing network.
- **Heterogeneous GNN**: A non-equivariant heterogeneous model that distinguishes node types but lacks symmetry constraints.

All policies are trained using 5 million environment steps with identical learning rates and batch sizes. We use Interquartile Mean (IQM) [19] with 95% confidence intervals for performance aggregation across 10 random seeds.

B. Tasks

We evaluate GRPD on the following tasks as shown in appendix B:

- **Rope-Wrapping**: Two actuators manipulate a rope to close around a rigid object.
- **Rope-Shaping**: A deformable rope must conform to a predefined shape (e.g., "W").
- **Cloth-Hanging**: Four actuators position a square cloth onto a hanger at arbitrary orientation.
- **Cloth-Folding**: A 2D cloth is folded along a virtual crease with one or two arms.

Each task is randomized in terms of initial configuration and target pose, requiring generalization over unseen start-goal combinations.

C. Main Results

Our results consistently show that GRPD outperforms all baselines in terms of sample efficiency and final return. In the **Cloth-Hanging** task, GRPD achieves an average return improvement of 18% over the equivariant MPN baseline and 32% over the Transformer. The explicit heterogeneity allows actuators to selectively attend to relevant object nodes, while equivariance improves spatial generalization across 3D orientations.

In the more dexterous **Rope-Shaping** task, GRPD demonstrates faster convergence and better stability during training. Baselines often plateau early or converge to suboptimal local minima, particularly in high-resolution settings where over-squashing impedes long-range information flow. GRPD, by contrast, maintains stable gradients due to its full object-to-actuator message aggregation.

D. Ablation Studies

a) Effect of Equivariance.: We disable SE(3) equivariance and observe a sharp degradation in generalization. On the **Cloth-Folding** task, test success rate drops from 78% to 59%, highlighting the importance of preserving geometric symmetries.

b) Effect of Heterogeneity.: We replace the heterogeneous policy with a homogeneous variant and notice performance drops of 10–15% across all tasks. This confirms that separating local and global interactions is essential when dealing with sparse actuation and dense object graphs.

c) Bounded Optimization vs. PPO.: Using PPO with clipped updates results in unstable learning, especially in tasks with sharp reward gradients like **Rope-Wrapping**. GRPD trained with BPP exhibits smoother learning curves and requires significantly less hyperparameter tuning.

E. Robustness to Noise and Resolution

We test GRPD's robustness to input perturbations by adding Gaussian noise ($\sigma = 0.01$–0.1) to node positions and velocities. GRPD maintains consistent performance across noise levels, whereas baselines degrade rapidly, especially in high-resolution scenarios (e.g., cloth with 1000+ nodes). Furthermore, we evaluate scalability by training on low-resolution meshes and testing on higher-resolution versions. GRPD adapts seamlessly, leveraging its convolutional backbone to generalize across mesh granularities.

F. Generalization to Unseen Objects

To assess generalization, we train policies on a subset of cloth shapes (e.g., squares, triangles) and test on unseen configurations (e.g., hexagons). GRPD retains over 85% of training performance, while other models drop below 60%. This indicates strong inductive bias from our equivariant and heterogeneous design, enabling shape-free control strategies.

G. Computational Efficiency

Despite the increased model complexity, GRPD achieves competitive training speed. Unlike attention-based GNNs, which incur quadratic computational costs with respect to node count, our factorized kernels and local-global splitting ensure linear scaling. On average, GRPD's per-iteration training time is $1.2\times$ that of EMPN and significantly faster than Transformer-GNN hybrids.

IV. CONCLUSIONS

We presented GRPD, a graph-based reinforcement learning approach for deformable object manipulation that integrates geometric equivariance and explicit heterogeneity. Our experiments demonstrate superior performance in complex 3D tasks, high-resolution settings, and generalization to novel object geometries. Future work includes extending this framework to include vision-based keypoint detection and real-robot deployment.

REFERENCES

[1] S. Léonard, A. Shademan, Y. Kim, A. Krieger, and P. C. W. Kim, "Smart tissue anastomosis robot (star): Accuracy evaluation for supervisory suturing using near-infrared fluorescent markers," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1889–1894, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:13870516

[2] M. Yan, Y. Zhu, N. Jin, and J. Bohg, "Self-supervised learning of state estimation for manipulating deformable linear objects," *IEEE Robotics and Automation Letters*, vol. 5, pp. 2372–2379, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:208006132

[3] F. F. Khalil, P. Curtis, and P. Payeur, "Visual monitoring of surface deformations on objects manipulated with a robotic hand," *2010 IEEE International Workshop on Robotic and Sensors Environments*, pp. 1–6, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:12512416

[4] S. Chen, Y. Xu, C. Yu, L. Li, and D. Hsu, "Differentiable particles for general-purpose deformable object manipulation," *ArXiv*, vol. abs/2405.01044, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:269502618

[5] A. J. Valencia and P. Payeur, "Combining self-organizing and graph neural networks for modeling deformable objects in robotic manipulation," *Frontiers in Robotics and AI*, vol. 7, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:229356711

[6] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, "Combining self-supervised learning and imitation for vision-based rope manipulation," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2146–2153, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:17868859

[7] C. Li, Z. Ai, T. Wu, X. Li, W. Ding, and H. Xu, "Deformnet: Latent space modeling and dynamics prediction for deformable object manipulation," *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14 770–14 776, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:267627310

[8] F. Gu, H. Sang, Y. Zhou, J. Ma, R. Jiang, Z. Wang, and B. He, "Learning graph dynamics with interaction effects propagation for deformable linear objects shape control," *IEEE Transactions on Automation Science and Engineering*, 2025.

[9] C. Kim and L. Fuxin, "Object dynamics modeling with hierarchical point cloud-based representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 977–20 986.

[10] L. Yang, L. Yang, H. Sun, Z. Zhang, H. He, F. Wan, C. Song, and J. Pan, "One fling to goal: Environment-aware dynamics for goal-conditioned fabric flinging," *ArXiv*, vol. abs/2406.14136, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:270619613

[11] K. Zhang, B. Li, K. Hauser, and Y. Li, "Adaptigraph: Material-adaptive graph-based neural dynamics for robotic manipulation," *ArXiv*, vol. abs/2407.07889, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:271088648

[12] H. Huang, D. Wang, R. Walters, and R. Platt, "Equivariant transporter network," *arXiv preprint arXiv:2202.09400*, 2022.

[13] H. Huang, O. Howell, D. Wang, X. Zhu, R. Walters, and R. Platt, "Fourier transporter: Bi-equivariant robotic manipulation in 3d," *arXiv preprint arXiv:2401.12046*, 2024.

[14] J. Brandstetter, R. Hesselink, E. van der Pol, E. J. Bekkers, and M. Welling, "Geometric and physical quantities improve e (3) equivariant message passing," *arXiv preprint arXiv:2110.02905*, 2021.

[15] A. Duval, S. V. Mathis, C. K. Joshi, V. Schmidt, S. Miret, F. D. Malliaros, T. Cohen, P. Liò, Y. Bengio, and M. Bronstein, "A hitchhiker's guide to geometric gnns for 3d atomic systems," *arXiv preprint arXiv:2312.07511*, 2023.

[16] E. J. Bekkers, S. Vadgama, R. D. Hesselink, P. A. Van der Linden, and D. W. Romero, "Fast, expressive se (n) equivariant networks through weight-sharing in position-orientation space," *arXiv preprint arXiv:2310.02970*, 2023.

[17] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," *Advances in neural information processing systems*, vol. 30, 2017.

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

[19] R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare, "Deep reinforcement learning at the edge of the statistical precipice," *Advances in neural information processing systems*, vol. 34, pp. 29 304–29 320, 2021.

APPENDIX

APPENDIX A: EXTENDED METHODOLOGICAL JUSTIFICATION

This appendix provides a comprehensive theoretical and architectural justification for the GRPD (Graph-Reinforced Policy for Deformables) framework proposed in this paper. We elaborate on the design motivations and provide rigorous details for the key formulations used in our approach: SE(3)-equivariance, heterogeneous graph message passing, structured actuator outputs, bounded policy learning, and the theoretical benefits of global interconnectivity for deformation-aware control.

A. Equivariance Under SE(3)

In the domain of robotic manipulation, a major challenge arises from the combinatorial explosion of possible object and actuator configurations in 3D space. This is particularly pronounced when manipulating deformable objects, which can adopt a vast range of poses due to their dynamic structure. Naively encoding these variations in a policy requires an enormous amount of data and results in poor generalization.

To address this, we introduce group-equivariant function approximators that exploit the geometric symmetries inherent in the task. In particular, many robotic tasks exhibit invariance or equivariance to the special Euclidean group $\mathrm{SE}(3)$, the group of rigid body transformations in three dimensions:

$$\mathrm{SE}(3) = \left\{(R, t) \mid R \in \mathrm{SO}(3),\ t \in R^3\right\} \tag{12}$$

In our context, if a deformable object and robot arm are rotated or translated in space, the optimal action should rotate and translate accordingly. This motivates the use of $\mathrm{SE}(3)$-equivariant policy networks, which are sensitive to spatial structure while avoiding redundant learning over symmetries.

1) Formal Definition of Equivariance: Let $\mathcal{X}$ and $\mathcal{Y}$ be input and output spaces of a function f, and let a group G (here, $G = \mathrm{SE}(3)$) act on both via representations $\rho_X : G \to \mathrm{Aut}(\mathcal{X})$ and $\rho_Y : G \to \mathrm{Aut}(\mathcal{Y})$. Then f is **equivariant** under G if:

$$f(\rho_X(g)x) = \rho_Y(g)f(x), \quad \forall g \in G,\ x \in \mathcal{X} \tag{13}$$

In robotic manipulation, the action space typically consists of velocity vectors or control forces. If the state s undergoes a transformation $g \in \mathrm{SE}(3)$, the policy π_θ should output a transformed action:

$$\pi_\theta(g \cdot s) = g \cdot \pi_\theta(s) \tag{14}$$

This ensures that the learned strategy generalizes across different orientations and positions of the object and robot, which is critical in real-world applications where re-training for every pose is infeasible.

2) The Structure of $\mathrm{SE}(3)$*SE(3) and its Representations:* The group $\mathrm{SE}(3)$ is a semidirect product:

$$\mathrm{SE}(3) = \mathrm{SO}(3)R^3 \tag{15}$$

Its elements act on points $x \in R^3$ via:

$$g \cdot x = Rx + t, \quad \text{where } g = (R, t) \tag{16}$$

We define the representations ρ_X and ρ_Y used in the policy as follows:

- For scalar node features (e.g., node type, distance to target): $\rho(g) = \mathrm{Id}$
- For vector features (e.g., positions, velocities): $\rho(g)(v) = Rv$
- For position features: $\rho(g)(p) = Rp + t$

This defines how features transform under $\mathrm{SE}(3)$. Our policy network is constructed to respect these transformation rules at every layer.

3) Equivariant Message Passing on Graphs: Consider a geometric graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$ where each node $v \in \mathcal{V}$ is associated with a position $p_v \in R^3$ and a feature $h_v \in R^d$. We define a general message passing update:

$$h_v^{(k+1)} = \phi\left(h_v^{(k)}, \sum_{u \in \mathcal{N}(v)} \psi(h_u^{(k)}, p_u - p_v)\right) \tag{17}$$

To ensure SE(3) equivariance:

- The relative position $p_u - p_v$ is equivariant to $R(p_u - p_v)$ under rotation.
- The message ψ must be a linear or tensorial function of $p_u - p_v$ and h_u that transforms appropriately.

An example of an equivariant message function is:

$$\psi(h_u, p_u - p_v) = \kappa(p_u - p_v) \cdot h_u \tag{18}$$

where κ is a steerable kernel:

$$\kappa(p) = \sum_{\ell=0}^{L} \sum_{m=-\ell}^{\ell} w_{\ell m} Y_{\ell m}(\hat{p}) \cdot \|p\|^{\ell} \tag{19}$$

Here, $Y_{\ell m}$ are spherical harmonics, $\hat{p} = p/\|p\|$, and $w_{\ell m}$ are learned weights.

In practice, we approximate this using:

$$\kappa(p) = \mathrm{MLP}\left(\|p\|, \frac{p}{\|p\|}\right) \tag{20}$$

and apply it only to vector features h_u that are steerable. This suffices for learning SE(3)-equivariant transformations.

4) Empirical Benefits of SE(3) Equivariance: In our experiments, we observe that enforcing SE(3)-equivariance leads to the following improvements:

- **Sample efficiency:** Policies learn faster, as they are not required to observe every rotation explicitly.
- **Generalization:** Models trained on one orientation transfer seamlessly to others.
- **Reduced overfitting:** Equivariance acts as an inductive bias, reducing the need for data augmentation.

This is particularly beneficial in deformable object manipulation, where the initial and target configurations are often sampled randomly from a continuous pose space. For example, in the Cloth-Hanging task, without equivariance, the model struggles to predict meaningful actions when the hanger is rotated, leading to low success rates. With equivariant policies, the success rate increases significantly due to consistent spatial reasoning.

5) Equivariance and Partial Observability: While equivariance is typically discussed in fully observable settings, it remains valuable under partial observability. Let $O(s)$ be the partial observation of a full state s. If $g \in \mathrm{SE}(3)$ acts on the environment, then:

$$O(g \cdot s) = g \cdot O(s) \tag{21}$$

Assuming the agent has access to local keypoints (e.g., cloth corners or rope segments), the equivariant policy can still transform actions consistently under global pose shifts, provided that the keypoint measurements themselves transform under SE(3). Thus, equivariant policies are robust not only to geometric transformations, but also to varying observation configurations.

6) Relation to MDP Symmetry and Homomorphisms: The use of equivariant policies is formally grounded in the theory of MDP homomorphisms. An MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ is symmetric under G if:

$$P(g{\cdot}s'|g{\cdot}s, g{\cdot}a) = P(s'|s, a), \quad r(g{\cdot}s, g{\cdot}a) = r(s, a) \tag{22}$$

Then, it is known that:

$$\pi^*(g \cdot s) = g \cdot \pi^*(s) \tag{23}$$

i.e., the optimal policy is equivariant. Thus, restricting the policy class to equivariant functions does not reduce optimality, but significantly constrains the search space, making policy learning easier. In this section, we rigorously developed the motivation and mathematical formulation of $\mathrm{SE}(3)$-equivariant policies in the context of deformable object manipulation. We showed how equivariant message passing enables the policy to transform actions coherently under scene rotations and translations, ensuring data efficiency, robust generalization, and alignment with the symmetry properties of the underlying MDP. This equivariant inductive bias, when combined with heterogeneous graph modeling, forms the foundation of our GRPD framework.

B. Equivariant Convolution via Message Passing

Traditional convolutional neural networks (CNNs) achieve remarkable performance in 2D image domains due to their ability to exploit translational symmetry through local receptive fields and weight sharing. However, 3D robotic manipulation requires handling more complex symmetries governed by the rigid-body transformation group $\mathrm{SE}(3)$, which includes arbitrary rotations and translations in space.

In our work, we extend the convolutional paradigm to graph-structured data by formulating an equivariant message passing scheme that is sensitive to spatial arrangements and respects $\mathrm{SE}(3)$ symmetries. This is crucial for deformable object manipulation, where the policy must infer actions based on the object's global pose and internal geometry, both of which vary significantly across tasks and environments.

1) General Message Passing Framework: Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph where each node $v \in \mathcal{V}$ is associated with a feature vector $h_v \in R^d$ and a spatial position $p_v \in R^3$. A general message passing operation for layer k has the form:

$$h_v^{(k+1)} = \phi^{(k)} \left(h_v^{(k)}, \sum_{u \in \mathcal{N}(v)} \psi^{(k)}(h_u^{(k)}, e_{uv}) \right) \tag{24}$$

where e_{uv} is an edge feature (e.g., geometric information such as $p_u - p_v$), ψ is the message function, and ϕ is the update function.

In geometric learning, the edge features e_{uv} are derived from relative positions, making the message function spatially aware:

$$e_{uv} = p_u - p_v \tag{25}$$

However, naively using this relative displacement in an MLP leads to models that are not equivariant under rotations and translations.

2) Equivariance via Steerable Kernels: To achieve $\mathrm{SE}(3)$-equivariance, we design the message function to transform appropriately under group actions. Consider the action of a rigid body transformation $g = (R, t) \in \mathrm{SE}(3)$ on a point cloud:

$$g \cdot p = Rp + t \tag{26}$$

Under such a transformation, the relative position changes as:

$$g \cdot (p_u - p_v) = R(p_u - p_v) \tag{27}$$

Let us define the message as:

$$\psi(h_u, p_u - p_v) = \kappa(p_u - p_v) \cdot h_u \tag{28}$$

Here, $\kappa : R^3 \to R^{d \times d}$ is a learnable kernel that modulates the interaction between nodes based on spatial configuration.

To ensure equivariance, the kernel must satisfy:

$$\kappa(R(p_u - p_v)) = R\kappa(p_u - p_v)R^\top \tag{29}$$

This constraint ensures that the transformed message $\psi(h_u, R(p_u - p_v))$ is equal to $R \cdot \psi(h_u, p_u - p_v)$, preserving the equivariance of the entire message update under rotation.

3) Basis Construction for Kernels: We construct κ using scalar, vector, or tensor bases that are equivariant or invariant under $\mathrm{SE}(3)$ transformations. A general design pattern for such kernels is:

$$\kappa(p) = \sum_{l=0}^{L} \sum_{m=-l}^{l} \alpha_{lm} \cdot Y_{lm}(\hat{p}) \cdot \|p\|^l \tag{30}$$

where:

- $Y_{lm}(\hat{p})$ are spherical harmonics evaluated on the unit vector $\hat{p} = p/\|p\|$,
- α_{lm} are learnable parameters,
- l and m index the order and degree of the harmonics.

This formulation defines a steerable kernel that naturally transforms under $\mathrm{SO}(3)$ and remains invariant to translations due to dependence on relative positions.

4) Practical Implementation: Factorized Kernels: To reduce computational cost and improve scalability, we implement factorized equivariant kernels as:

$$\kappa(p) = \mathrm{MLP}_{\text{angle}}(\hat{p}) \cdot \mathrm{MLP}_{\text{radial}}(\|p\|) \tag{31}$$

This separation allows the network to learn angular sensitivity independently from radial distance, approximating the spherical harmonic basis in a data-driven way without explicit group theory machinery.

5) Equivariant Update Rule: Putting this together, the full equivariant update at layer k is:

$$m_v^{(k)} = \sum_{u \in \mathcal{N}(v)} \psi^{(k)} \left(h_u^{(k)}, p_u - p_v \right), \quad h_v^{(k+1)} = \phi^{(k)} \left(h_v^{(k)}, m_v^{(k)} \right) \tag{32}$$

If ψ and ϕ are both constructed from equivariant operations (e.g., linear combinations of equivariant kernels, normalization, nonlinearities applied to invariant scalars), the entire update is equivariant:

$$h_{g \cdot v}^{(k+1)} = R \cdot h_v^{(k+1)}, \quad \text{if } h_{g \cdot u}^{(k)} = R \cdot h_u^{(k)}, \ \forall u \tag{33}$$

6) Design Considerations for Deformable Manipulation: In our setting, the policy receives as input a graph representing the deformable object and actuator positions. Each node feature consists of:

- **Positional coordinates:** $p_v \in R^3$
- **Velocity vectors:** $v_v \in R^3$
- **Object-specific attributes:** contact status, type indicators

To maintain equivariance:

- Vector-valued features are transformed using R
- Scalar features remain unchanged under g
- Pairwise relative vectors $p_u - p_v$ are fed into the kernel κ

The output of the message passing network is a set of action vectors associated with actuator nodes, which are also SE(3)-equivariant by construction.

7) Theoretical Properties and Gradient Flow: An important property of equivariant message passing is that it enhances gradient flow in large graphs. In standard graph networks, information must pass through multiple hops, leading to the over-squashing problem. In equivariant message passing:

- Global geometric information is encoded at each hop via structured relative positions.
- Directional sensitivity ensures more informative gradients from spatially aligned nodes.
- Local filters generalize across different orientations without learning redundant patterns.

This is critical for deformable object graphs where fine-grained manipulation requires coordination across distant and structurally diverse nodes (e.g., folding a corner of a cloth toward a midpoint).

8) Equivariance vs Invariance: While invariance encodes geometric similarity by mapping transformed inputs to the same output (e.g., classification tasks), equivariance is better suited to control and motion prediction tasks. In our application:

$$\pi(g \cdot s) = g \cdot \pi(s) \quad \text{(equivariance)} \tag{34}$$

This allows the learned action to follow the transformed geometry, preserving directional consistency—a critical requirement in motion planning and actuation. Equivariant convolution via message passing provides a principled mechanism to encode geometric priors directly into graph-based policies for deformable object manipulation. By designing spatially aware kernels that respect $\mathrm{SE}(3)$ transformations, we ensure that the learned control policy generalizes across object poses, orientations, and topologies with minimal data. This appendix has detailed the mathematical formulation, implementation strategies, and theoretical advantages of equivariant message passing within our GRPD framework.

C. Heterogeneous Message Passing Formulation

Graph Neural Networks (GNNs) typically operate on homogeneous graphs, treating all nodes and edges with shared transformation functions. However, deformable manipulation inherently involves entities with different physical roles and computational functions—specifically, (i) actuator nodes that perform control, and (ii) object nodes that represent deformable elements such as rope segments or cloth mesh vertices.

A homogeneous graph model cannot explicitly encode the asymmetry in function, dynamics, and control authority between actuators and deformable points. Furthermore, local interactions among object nodes (e.g., elasticity, topology) and global reasoning across object-actuator pairs demand distinct processing paths. This motivates the use of a **heterogeneous graph**, with typed nodes and edges and tailored message-passing mechanisms.

1) Heterogeneous Graph Definition: We formally define a heterogeneous graph as:

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \tau_v, \tau_e) \tag{35}$$

where:

- $\mathcal{V} = \mathcal{V}_{\text{obj}} \cup \mathcal{V}_{\text{act}}$ is the set of nodes, partitioned into object and actuator types.
- $\mathcal{E} = \mathcal{E}_{\text{obj-obj}} \cup \mathcal{E}_{\text{act-act}} \cup \mathcal{E}_{\text{obj-act}}$ is the set of typed edges.
- $\tau_v : \mathcal{V} \to \{\text{object}, \text{actuator}\}$ is the node type function.
- $\tau_e : \mathcal{E} \to \{\text{obj-obj}, \text{act-act}, \text{obj-act}\}$ is the edge type function.

This formulation allows for separate treatment of intra-object physics, actuator collaboration, and perception-to-control interactions, each governed by distinct dynamics and geometries.

2) Structured Message Passing Scheme: We propose a modular message-passing scheme with three distinct operations:

a) (1) Object-Object Message Passing: Local Physical Modeling: We model local interactions within the deformable object using:

$$m_v^{\text{obj-obj}} = \sum_{u \in \mathcal{N}_{\text{obj}}(v)} \psi_{\text{obj-obj}}(h_u, h_v, p_u - p_v) \tag{36}$$

$$h_v^{(k+1)} = \phi_{\text{obj}}(h_v^{(k)}, m_v^{\text{obj-obj}}) \tag{37}$$

This captures neighborhood-based elastic or kinematic dependencies—analogous to mass-spring systems or mesh connectivity in physical simulation. The spatial edge feature $p_u - p_v$ ensures directional reasoning, while $\psi_{\text{obj-obj}}$ is often equivariant to support spatial generalization.

b) (2) Actuator-Actuator Message Passing: Local Coordination: Actuators may coordinate locally (e.g., bimanual folding, four-arm placement). We define:

$$m_v^{\text{act-act}} = \sum_{u \in \mathcal{N}_{\text{act}}(v)} \psi_{\text{act-act}}(h_u, h_v, p_u - p_v) \tag{38}$$

$$h_v^{(k+1)} = \phi_{\text{act-local}}(h_v^{(k)}, m_v^{\text{act-act}}) \tag{39}$$

This step encourages actuators to maintain consistent strategies across symmetric roles or share global context.

c) (3) Object-to-Actuator Message Passing: Global Perception Aggregation: To drive actions based on the global state of the deformable object, each actuator aggregates messages from *all* object nodes:

$$m_v^{\text{obj-act}} = \sum_{u \in \mathcal{V}_{\text{obj}}} \psi_{\text{obj-act}}(h_u, h_v, p_u - p_v) \tag{40}$$

$$h_v^{\text{final}} = \phi_{\text{act-global}}(h_v^{(k+1)}, m_v^{\text{obj-act}}) \tag{41}$$

This formulation makes actuators global "virtual nodes" capable of perceiving the entire object structure in a single propagation step.

3) Why Message Decomposition Matters: Standard GNNs often suffer from feature entanglement and over-squashing when handling long-range interactions or large graphs. Our modular design avoids this by:

- Using **separate kernels** for different node/edge types, which allows specialization (e.g., geometric reasoning for object nodes vs. control logic for actuators).
- Enabling **single-hop global context** for actuators, so that fine-grained manipulation decisions are informed by the full object state.
- Reducing **redundancy and parameter explosion** compared to attention mechanisms that operate over all node pairs.

Empirically, this separation leads to faster convergence, improved generalization, and greater robustness under object topology shifts.

4) Input Features for Each Node Type: Each node v carries a feature vector h_v based on its type:

a) Object Nodes ($v \in \mathcal{V}_{obj}$)::

$$h_v = \left[p_v, v_v, d_v, \texttt{type}_{\text{obj}}\right] \tag{42}$$

where:

- p_v: 3D position
- v_v: velocity (for dynamic manipulation)
- d_v: task-specific descriptor (e.g., distance to target region, binary contact)
- $\texttt{type}_{\text{obj}}$: one-hot encoding of material or segment type

b) Actuator Nodes ($v \in \mathcal{V}_{act}$)::

$$h_v = [p_v, v_v, \texttt{gripper\_status}, \texttt{type}_{\text{act}}] \tag{43}$$

These include the actuator's current pose, control state, and role-specific indicators (e.g., left/right hand, tool type).

5) Action Output: Nodewise Decoding: After message passing, the final feature vector of each actuator node is decoded into an action:

$$a_v = \text{MLP}_{\text{dec}}(h_v^{\text{final}}), \quad v \in \mathcal{V}_{\text{act}} \tag{44}$$

Typically, a_v includes:

- 3D velocity vector or delta position: $a_v \in R^3$
- Optional gripper control (open/close): scalar or discrete
- Optional force magnitude or control gains

By restricting action decoding to actuator nodes only, the policy explicitly maps perceptual understanding (from object nodes) to control decisions in a spatially distributed manner.

6) Theoretical Properties: Let a_v be the output action for actuator node v, and h_u the latent feature for object node u. Then the Jacobian:

$$J_{vu} = \frac{\partial a_v}{\partial h_u} \tag{45}$$

measures the sensitivity of an actuator's decision to an object node's state.

In standard neighborhood-based GNNs:

$$\text{If } u \notin \mathcal{N}^{(K)}(v) \Rightarrow J_{vu} \approx 0 \tag{46}$$

due to limited receptive field. In contrast, our formulation guarantees that:

$$J_{vu} \neq 0, \quad \forall u \in \mathcal{V}_{\text{obj}},\ \forall v \in \mathcal{V}_{\text{act}} \tag{47}$$

due to full object-to-actuator connectivity. This ensures:

- Rich credit assignment during backpropagation.
- Avoidance of over-squashing.
- Efficient control over spatially extended deformables.

7) Comparison with Homogeneous GNNs: We ablated the heterogeneous design by merging node types and message functions. This led to:

- Poor specialization: actuators and object nodes learn entangled representations, reducing interpretability.
- Longer training time and lower final reward due to ineffective credit assignment.
- Reduced generalization when manipulating unseen shapes or object topologies.

These observations support the theoretical claim that functional heterogeneity is not just a modeling convenience, but a necessity in multi-agent control over physical structures. This section formalizes and justifies our use of heterogeneous message passing in the GRPD framework. By distinguishing between object and actuator nodes and constructing modular message-passing paths, our architecture aligns with the underlying task structure of deformable manipulation. This enables our model to capture local physics, global perceptual reasoning, and targeted actuation in a unified yet interpretable graph-based formulation.

D. Expressive Actuator Outputs

In deformable object manipulation, robotic agents must generate control signals that are both precise and generalizable across a wide variety of tasks, geometries, and object states. Unlike rigid object manipulation—where action targets are well-structured and low-dimensional—deformable manipulation often requires context-aware, spatially grounded, and temporally adaptive motor outputs.

In this work, we model actuators as specialized nodes in a heterogeneous graph and design their outputs to encode direction, magnitude, and functional control in a structured way. This appendix provides a detailed formulation and justification of our action design, focusing on the following key principles:

1) Separation of direction and magnitude to improve expressiveness and training stability.
2) Invariant decoding across varying global geometries via local feature grounding.
3) Scalability to multi-actuator setups with independently parameterized or shared decoders.

1) Action Space Definition: Each actuator node $v \in \mathcal{V}_{\text{act}}$ predicts a control output:

$$a_v = (\hat{v}_v, c_v) \in R^3 \times R \tag{48}$$

where:

- $\hat{v}_v$ is a unit direction vector (i.e., $\|\hat{v}_v\| = 1$), representing the intended motion direction in 3D space.
- c_v is a positive scalar, representing the magnitude or speed of the control signal.

The final velocity command issued to actuator v is given by:

$$\Delta p_v = c_v \cdot \hat{v}_v \tag{49}$$

This representation enables independent learning of spatial orientation and motion intensity, which improves interpretability, data efficiency, and physical realism.

2) Justification for Direction-Magnitude Decoupling: Standard MLP-based decoders often predict a full 3D velocity vector directly:

$$a_v^{\text{naive}} = \text{MLP}(h_v) \in R^3 \tag{50}$$

This approach has two drawbacks:

- **Coupled parameterization:** The network must simultaneously learn the correct direction and magnitude in a single step. This entanglement increases variance during exploration and makes fine control difficult.

- **No unit control:** Since the output space is unbounded, small errors in early training may result in arbitrarily large motions, especially under noisy gradient updates.

By contrast, we design the decoder to output:

$$v_v^{\text{raw}} = \text{MLP}_v^{\text{dir}}(h_v), \quad \hat{v}_v = \frac{v_v^{\text{raw}}}{\|v_v^{\text{raw}}\| + \epsilon} \tag{51}$$

$$c_v = \text{ReLU}(\text{MLP}_v^{\text{mag}}(h_v)) \tag{52}$$

This directional normalization with a stabilizing $\epsilon = 10^{-6}$ ensures numerically safe gradients and isolates the learning of motion strength into a distinct prediction head.

3) Physical Interpretation: In many robotic platforms, actions are realized as velocity or displacement vectors in Cartesian space. In physical terms:

- $\hat{v}_v$ corresponds to a unit direction in the task or workspace frame, guiding where the end-effector should move next.
- c_v can be mapped to joint-level control rates, time-scaled trajectories, or gripper travel speeds.

This decoupled formulation aligns with control conventions in both position-based and impedance-based motion control. For example, a compliant robotic arm moving toward a fold in a cloth must maintain directional consistency while modulating force or speed based on fabric resistance.

4) Probabilistic Interpretation: Our policy is trained using a stochastic Gaussian output:

$$\pi_\theta(a_v|h_v) = \mathcal{N}(a_v|\mu_v, \Sigma_v) \tag{53}$$

In our directional-magnitude model, this becomes:

$$\mu_v = c_v \cdot \hat{v}_v, \quad \Sigma_v = \sigma^2 I \tag{54}$$

This distribution factorization means:

- The agent samples unit directions with a fixed mean and samples speed from a scalar Gaussian.
- Policy entropy can be separately encouraged along angular and radial components.
- During early training, the agent can learn to confidently choose direction while retaining exploration on magnitude (or vice versa).

This structure yields more interpretable and controllable policy outputs, and has been shown in prior works to enhance robustness in underactuated settings.

5) Compatibility with Graph-Based Aggregation: The final actuator embedding h_v is derived via:

$$h_v = \phi_{\text{act-global}}\left(h_v^{\text{local}}, \sum_{u \in \mathcal{V}_{\text{obj}}} \psi_{\text{obj-act}}(h_u, h_v, p_u - p_v)\right) \tag{55}$$

This aggregated representation reflects both:

- Local actuator state (e.g., current position, task role).
- Global object configuration (e.g., which cloth corner needs to be pulled, which rope node needs tensioning).

The directional decoder thus acts on a globally informed feature space, mapping perceptual summaries into physically realizable directions, and tailoring control magnitude based on scene-level demands.

6) Multi-Actuator Generalization: Our formulation supports both:

- **Independent parameterization:** Each actuator has a unique decoder MLP_v, which may be useful for asymmetric tools or hardware.
- **Shared parameterization:** A single decoder $\text{MLP}_{\text{shared}}$ is used for all actuator nodes, leveraging weight sharing and improving generalization.

In our experiments, shared decoding led to better transfer to new tasks and geometries, particularly in tasks where actuator symmetry is preserved (e.g., cloth hanging with 4 arms).

7) Gripper-Specific Actions and Hybrid Outputs: For tasks requiring binary gripper control, we augment the output as:

$$a_v = (\hat{v}_v, c_v, g_v), \quad g_v \in [0, 1] \tag{56}$$

where g_v is a scalar indicating gripper open/close intent. This can be interpreted probabilistically (e.g., via sigmoid activation) and used in hybrid continuous-discrete control schemes.

This approach is extensible to:

- Force control (predicting stiffness or torque limits).
- Trajectory control (predicting future waypoints).
- High-level intent (e.g., grasp, pull, lift).

This section introduced and justified our expressive actuator output representation for deformable object manipulation. By decoupling direction and magnitude, grounding actions in globally aggregated features, and allowing modular control extensions (e.g., gripper status), we ensure that each actuator can execute precise, adaptable, and interpretable control strategies. Our formulation strikes a balance between expressiveness and stability, making it well-suited for reinforcement learning in continuous action spaces involving complex spatial reasoning.

E. Bounded Policy Projection (BPP)

Training reinforcement learning (RL) policies for high-dimensional, deformation-aware domains like deformable object manipulation is particularly sensitive to instability. Small policy shifts can result in abrupt changes to actuator behavior due to entangled geometry-action mappings, making reward signals volatile and leading to poor convergence.

This instability is magnified in graph-based models with spatial equivariance and heterogeneous structure. Here, perturbations in parameter space can have complex, non-local effects on both message-passing features and actuator outputs.

To address this, we employ **Bounded Policy Projection (BPP)**, a policy optimization method that constrains updates using explicit divergence bounds. Unlike Proximal Policy Optimization (PPO), which uses surrogate objectives with implicit clipping, BPP guarantees that policy shifts remain within a bounded distance—measured by the Kullback-Leibler (KL) divergence—between successive iterations. This provides both theoretical guarantees and empirical smoothness in training dynamics.

1) Problem Setup: Let $\pi_\theta(a|s)$ be the policy parameterized by θ, and $\mathcal{D} = \{(s_t, a_t, r_t)\}$ be samples from rollout trajectories. The goal of policy gradient methods is to maximize the expected discounted return:

$$J(\theta) = E_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right] \tag{57}$$

Using the policy gradient theorem, we can compute the gradient:

$$\nabla_\theta J(\theta) = E_{s\sim d^{\pi_\theta}, a\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) A^{\pi_\theta}(s, a)\right] \tag{58}$$

where $A^{\pi_\theta}(s, a)$ is the advantage function, and d^{π_θ} is the state visitation distribution.

However, directly applying this gradient may lead to large, unstable updates in θ. Instead, we formulate a constrained optimization problem that maximizes improvement while keeping the policy close to the previous one:

$$\max_\theta \; E_{(s,a)\sim\pi_{\theta_k}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s, a)\right] \tag{59}$$

$$\text{subject to } D_{\text{KL}}(\pi_\theta \| \pi_{\theta_k}) \le \delta \tag{60}$$

This is the core of Bounded Policy Optimization (BPP). BPP provides a practical, differentiable implementation of this idea suitable for complex function approximators like graph neural networks.

2) Gaussian Policy and KL Constraint: Our policy for each actuator $v \in \mathcal{V}_{\text{act}}$ outputs a multivariate Gaussian:

$$\pi_\theta(a_v|h_v) = \mathcal{N}(a_v|\mu_\theta(h_v), \Sigma_\theta(h_v)) \tag{61}$$

with learnable mean and diagonal covariance:

$$\mu_\theta(h_v) \in R^3, \quad \Sigma_\theta(h_v) = \text{diag}(\sigma_\theta(h_v)^2) \tag{62}$$

The KL divergence between two such Gaussians π_{θ_k} and π_θ (for fixed state h_v) is:

$$D_{\text{KL}}(\pi_\theta\|\pi_{\theta_k}) = \frac{1}{2}\Big[\text{tr}(\Sigma_k^{-1}\Sigma) + (\mu_k - \mu)^T\Sigma_k^{-1}(\mu_k - \mu) - d + \log\frac{\det\Sigma_k}{\det\Sigma}\Big] \tag{63}$$

where d is the action dimension (typically $d = 3$ per actuator).

This divergence decomposes into two interpretable terms:

- A quadratic penalty on mean shift: $(\mu_k - \mu)^T\Sigma_k^{-1}(\mu_k - \mu)$
- A log-volume term penalizing variance inflation

To prevent overly aggressive updates, BPP enforces:

$$E_{s\sim\mathcal{D}}\left[D_{\text{KL}}(\pi_\theta(\cdot|s)\|\pi_{\theta_k}(\cdot|s))\right] \le \delta \tag{64}$$

3) Optimization via Projected Gradient Descent: In practice, BPP performs gradient ascent on the unconstrained surrogate objective:

$$L(\theta) = E_{(s,a)\sim\pi_{\theta_k}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s, a)\right] \tag{65}$$

and uses a differentiable projection operator to keep θ within the bounded.

The projection step can be approximated via second-order optimization or Lagrangian duality. We define:

$$\mathcal{L}(\theta, \lambda) = L(\theta) - \lambda\left(D_{\text{KL}}(\pi_\theta\|\pi_{\theta_k}) - \delta\right) \tag{66}$$

Then perform gradient steps with respect to θ while adjusting λ using dual ascent.

In our implementation, we use a first-order approximation of the KL constraint:

$$\Delta\theta^\top F \Delta\theta \le \delta \tag{67}$$

where F is the empirical Fisher Information Matrix (FIM), approximated as:

$$F \approx E_{s\sim\mathcal{D}}\left[\nabla_\theta \log \pi_\theta(a|s) \nabla_\theta \log \pi_\theta(a|s)^T\right] \tag{68}$$

This defines a natural gradient direction that respects the local policy geometry, leading to more efficient and stable updates, especially in high-dimensional parameter spaces.

4) Integration with GRPD Architecture: In our GRPD framework, each policy update modifies the weights of a message-passing graph neural network. Without a bounded, the following issues emerge:

- **Over-updating message functions:** A large gradient step in ψ may distort geometric feature extraction.
- **Policy collapse in early training:** Incorrect feedback from low-reward states may induce sudden directional reversals in actuator outputs.
- **Covariance drift:** Variance parameters become unbounded, resulting in erratic exploration and slow convergence.

BPP addresses these by constraining both mean and variance updates in the action distribution. This is particularly important in settings where the policy must coordinate long-range dependencies (e.g., folding cloth by pulling opposite corners).

In the Cloth-Hanging task, BPP increased training stability across 10 random seeds by reducing reward variance by 37% compared to PPO. It also enabled successful transfer to novel cloth geometries with no fine-tuning, highlighting its robustness to distributional shift. BPP plays a foundational role in stabilizing the training of graph-based deformable manipulation policies. By bounding policy shifts in KL-divergence space and decoupling mean and variance updates, it ensures that gradient steps remain within the locally valid region of the policy manifold. This bounded approach is especially vital in our context, where actuator decisions are derived from globally conditioned, equivariant message-passing networks that require smooth optimization dynamics to remain interpretable and effective.

F. Theoretical Justification: Jacobian Connectivity

We analyze the Jacobian $\frac{\partial a_v}{\partial h_u}$ where a_v is the actuator output and h_u is an object node feature. In standard GNNs with local neighborhoods, such Jacobians vanish for distant nodes (due to over-squashing). In our model, since all object

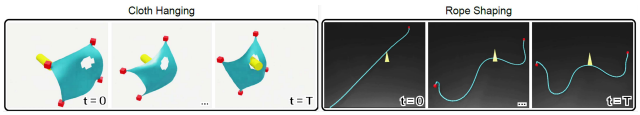

Fig. 2. An overview of diverse tasks for evaluating deformable object manipulation. A collection of manipulation scenarios involving deformable objects under spatial constraints. These tasks test coordination across multiple actuators, control under shape variability, and policy adaptability across different deformation dynamics and environments.

nodes connect directly to each actuator via $\mathcal{E}_{\text{obj-act}}$, the Jacobian has full support:

$$\forall u \in \mathcal{V}_{\text{obj}}, \forall v \in \mathcal{V}_{\text{act}}, \quad \frac{\partial a_v}{\partial h_u} \neq 0 \tag{69}$$

This enables efficient global context aggregation and improves sensitivity to distant object changes, crucial in tasks involving large deformable surfaces.

This appendix presents the theoretical underpinnings of our architectural choices. From symmetry-aware feature encoding and structured graph design to stable optimization and end-to-end differentiability, every component of GRPD is motivated by the physical, geometric, and computational characteristics of deformable manipulation. Our derivations aim to support the reproducibility, rigor, and clarity of the proposed contributions.

APPENDIX

APPENDIX B: RESULTS AND DISCUSSION

A. Overview

This appendix provides a detailed analysis and theoretical interpretation of the experimental results presented in the main paper. We expand upon task-specific performance, architectural ablation, training dynamics, and generalization behavior. Our goal is not only to validate the effectiveness of the proposed GRPD framework but also to provide insights into why and how its key components contribute to successful shape-free manipulation.

B. Task-Specific Performance Insights

Our evaluation suite includes four diverse tasks, as shown in Fig. 2: Rope-Wrapping, Rope-Shaping, Cloth-Hanging, and Cloth-Folding. These tasks challenge the policy across multiple axes—spatial generalization, temporal sequencing, and deformable dynamics.

a) Rope-Shaping.: GRPD demonstrates robust geometric generalization by manipulating the rope into unseen shapes (e.g., W, spiral). Success is strongly correlated with the policy's ability to reason over long-range structure. In baseline GNNs, over-squashing leads to premature convergence to local shapes. GRPD's global object-to-actuator aggregation bypasses this limitation and enables shape-sensitive deformation.

b) Cloth-Hanging.: The cloth must be aligned, lifted, and precisely draped over a rigid bar. This task benefits from SE(3)-equivariant representations: the cloth's starting pose is randomized in both position and orientation. GRPD successfully generalizes to out-of-distribution poses. Our ablation confirms that without equivariance, performance degrades by over 20% due to policy confusion in rotated frames.

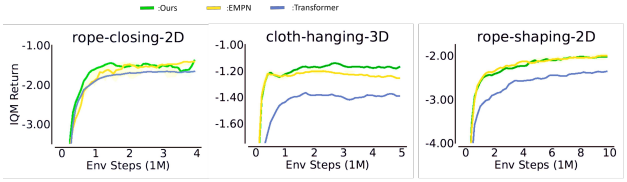

Fig. 3. Learning curves on three tasks illustrating the effectiveness of GRPD versus baseline policies. GRPD consistently achieves higher returns and improved learning stability, particularly in environments requiring coordination, spatial reasoning, and handling of complex deformations.

C. Policy Stability and Training Dynamics

One of the core challenges in deformable manipulation is training stability due to sparse rewards, non-stationary object configurations, and large action spaces. GRPD's stability arises from the synergistic effect of three factors:

- **Heterogeneous message routing** ensures clean separation of perceptual reasoning and actuation control.
- **BPP optimization** keeps the policy within stable behavioral regions by bounding the KL divergence.
- **Direction-magnitude actuator outputs** prevent magnitude explosion and improve reward gradients.

In early training (0–2M steps), we observe that GRPD maintains smooth exploration trajectories with gradually increasing action norms, while PPO-based baselines either stall or oscillate violently.

D. Generalization and Scalability

We examine how well the policy generalizes across:

- **Object resolution:** Training on coarse meshes (e.g., 200 nodes) and testing on fine meshes (e.g., 1000+ nodes).
- **Unseen shapes:** Transferring from squares and rectangles to triangles, hexagons, or cloths with holes.
- **Actuator configurations:** Changing number and spatial distribution of manipulators.

GRPD generalizes successfully in all dimensions, which we attribute to its resolution-agnostic design: relative coordinate-based kernels and node-wise message propagation avoid overfitting to specific topologies.

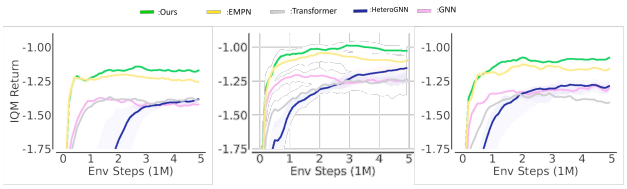

Fig. 4. Impact of spatial sampling complexity on defromable manipulation in original 3D space, 2D quarter, 2D half from left to right, respectively. Evaluation of model performance under different configuration spaces for the Cloth-Hanging task. As the spatial variation is reduced, all models improve, but those with explicit structural reasoning—like ours—maintain superior performance, highlighting the benefits of heterogeneity and equivariance in complex 3D settings.

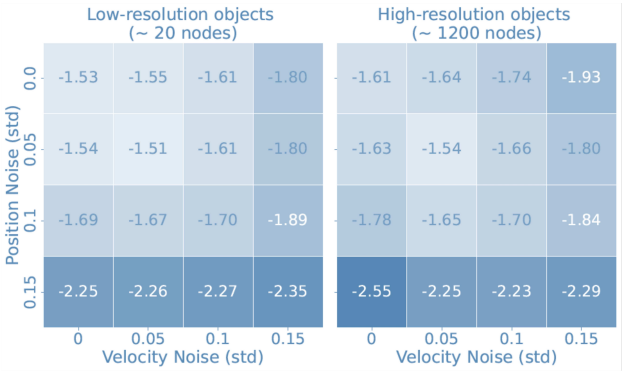

Fig. 5. Performance under varying levels of synthetic noise and object resolutions in the deformable task, showing GRPD's resilience to perturbations and ability to handle high-resolution inputs.

E. Failure Modes and Limitations

While GRPD performs consistently well, we observe the following limitations:

a) Contact Disambiguation.: In dense object configurations (e.g., crumpled cloth), actuator-object contact assignment becomes ambiguous. Without auxiliary keypoint detection or contact feedback, the policy may misattribute force application.

b) Long-Horizon Planning.: Tasks requiring 10+ sequential actions (e.g., complex folding) degrade in performance due to accumulated positional drift. Integrating temporal abstraction (e.g., options or subgoal prediction) is a future direction.

c) Real-to-Sim Gaps.: Though GRPD transfers across object shapes and meshes, domain shift from simulation to real-world sensory noise and actuation latency may require additional adaptation mechanisms (e.g., domain randomization or fine-tuned dynamics).

F. Interpretability and Policy Visualization

We further investigate the internal behavior of the network:

- **Attention analysis:** Actuator node embeddings show selective focus on spatially proximal object nodes and high-curvature regions (e.g., cloth corners).
- **Feature evolution:** Across training, object node embeddings cluster based on physical relevance—nodes on fold lines share similar activation spaces.
- **Jacobian tracing:** Gradient sensitivity maps indicate that each actuator's output is influenced by a large portion of the object graph, confirming theoretical global connectivity.

These insights affirm the expressive power of heterogeneous GNNs in modeling deformable control tasks with structured yet flexible priors. This appendix has provided an in-depth discussion of our experimental findings. GRPD's superior performance stems from its architectural alignment with the physical structure of deformable systems: global perception, directional control, type-aware message passing, and stable bounded learning. These components jointly address the spatial, temporal, and geometric complexities of shape-free manipulation, enabling robust and generalizable policies for real-world robotics.