Extended Abstract

Motivation Imitation learning offers a powerful framework for training robot manipulation policies. However, policies trained on a specific robot often fail to generalize across different morphologies. For example, a policy trained on a 7-DoF arm may not work on a 6-DoF arm due to differences in kinematics and control spaces. This limits scalability, as collecting new demonstrations for each platform is labor-intensive. To address this, we explore morphology-aware imitation learning to enable skill transfer across robots with varying structures—e.g., joint count, link length, actuation types—without retraining from scratch. This capability is essential for deploying learning-based policies in diverse, real-world robotic systems.

Method We investigate six approaches for generalizing imitation policies across different robot morphologies. As a baseline, **Per-Robot Policies (PRP)** train separate policies for each robot. **Direct Transfer (DT)** naively deploys a policy trained on one robot onto others without adaptation. In **Concatenated Morphology Vector (CMV)**, a fixed-length vector encoding robot morphology is appended to the observation input. **CMVM** adds a binary mask indicating valid joints to help the network distinguish padded dimensions. **GNN Morphology Embedding (GNN-ME)** uses a Graph Neural Network to compute an embedding from a graph representation of the robot's kinematic chain. Finally, **End-Effector Pose Policy + IK (EEPP)** trains a policy to predict end-effector displacements in task space, which are then converted into joint velocities using a robot-specific inverse kinematics solver.

Implementation We use the RLBench simulator to evaluate these methods on three robot arms: Franka Panda (7-DoF), UR5e (6-DoF), and KUKA iiwa-14 (7-DoF), across three tasks—Reach Target, Door Opening, and Basketball in Hoop. Each robot-task pair includes 100 demonstrations, totaling 900. Policies (except EEPP) take as input RGB images, joint states, and optionally morphology vectors, and output joint velocities using a Mixture Density Network (MDN). The MDN models a multimodal distribution over actions. Unified policies are trained on combined data from all robots; PRP and DT use data from one robot only. EEPP outputs end-effector deltas, converted via IK to joint velocities.

Results Quantitative results show PRP policies achieve the highest success (90–95%) on their respective robots. DT fails to generalize (5% success) due to mismatched control spaces. CMV enables moderate generalization (50–60%), with CMVM offering marginal gains. GNN-ME performs poorly (<10%), likely due to insufficient morphology diversity. EEPP achieves near-PRP performance across all robots and tasks, solving even the most complex (Basketball in Hoop) scenario. Qualitatively, EEPP and PRP consistently execute smooth, effective motions. CMV and CMVM occasionally succeed in simpler tasks but fail on complex ones. DT outputs erratic or static commands on unseen robots due to joint dimensional mismatch.

Discussion Our study highlights the effectiveness of morphology-aware conditioning. CMV improves transfer by allowing the policy to adapt its output based on robot attributes. The marginal impact of CMVM suggests the policy can ignore padded morphology entries without explicit masking. GNN-ME underperforms, underscoring that expressive encodings require diverse training data. EEPP succeeds by shifting control to end-effector space, avoiding direct dependence on robot-specific joint configurations. However, EEPP assumes access to accurate kinematic models and solvers, and it may struggle in tasks requiring dynamic interactions or contact forces not captured by pure kinematics.

Conclusion We present a comprehensive evaluation of morphology-aware imitation learning across heterogeneous robot morphologies. Our results show that while directly transferring policies across morphologically distinct robots fails, incorporating structural information (CMV) or delegating control to IK (EEPP) enables strong generalization. EEPP in particular achieves performance near that of per-robot experts. These findings point toward promising directions for scalable, transferable robot learning. Future work could explore hybrid approaches combining morphology encoding and task-space planning with online adaptation or fine-tuning, as well as expanding to more robots and dynamic tasks to further stress-test generalization.

Morphology-Aware Imitation Learning for Cross-Robot Generalization

Jermaine Zhao
Department of Computer Science
Stanford University
zx2004@stanford.edu

Abstract

Learning robot manipulation via imitation is often hindered by poor cross-robot generalization, as policies trained on one robot fail to transfer to others with different morphologies. We address this challenge by developing and evaluating morphology-aware imitation learning methods that enable policy transfer across robots with varying degrees of freedom and kinematic structures. Using a Mixture Density Network, we train a base policy on the 7-DoF Franka Panda robot and compare six transfer approaches—including morphology-conditioned networks and end-effector–based control via inverse kinematics—on three tasks (Reach, Door Opening, Basketball) and three robots (Panda, UR5e, KUKA iiwa). Results show that directly transferring policies fails, while conditioning on morphology improves generalization. Notably, our End-Effector Pose Policy approach achieves near-expert performance on all robots without retraining. This work demonstrates that morphology-aware designs, particularly task-space abstraction, can significantly reduce the need for per-robot data, enabling more scalable and flexible deployment of learned robotic skills.

1 Introduction

Robotic manipulation skills learned by imitation tend to be narrowly specialized to the robot on which they were trained. If a policy is trained (via behavior cloning or other imitation learning) to perform a task with one robot arm, it will often fail when deployed on a different robot arm that has even slight differences in morphology – for instance, a different number of joints, link lengths, or joint actuation limits. This problem severely limits the scalability of robot learning, as new morphologies are common in real-world scenarios and retraining a policy from scratch for each robot is time-consuming and costly. The goal of this project is to develop methods for cross-robot generalization in imitation learning, enabling a policy to transfer across embodiments. In other words, we aim to answer: How can we train robot manipulation policies that generalize across different robot morphologies and tasks, using only imitation learning?

There are several challenges in this setting. Robots with different kinematic structures will produce different joint trajectories to achieve the same end-effector behavior, so a policy conditioned only on raw observations (images, joint states) might not know how to adjust its outputs for a new robot. Moreover, a policy trained on demonstrations for one robot might output invalid or suboptimal actions on another robot (for example, controlling a non-existent joint). Naively transferring policies can result in catastrophic failures, as we confirm in our experiments. Our key insight is that providing the policy with explicit information about the robot's morphology (or otherwise designing the policy to be morphology-agnostic) can enable generalization without retraining from scratch. We explore this insight through multiple approaches, from simple feature augmentation to more structured representations.

2 Related Work

Adapting control policies across robot morphologies has been studied through both online and offline approaches. Online adaptation methods like DEFT fine-tune policies on new tasks or environments using human demonstrations and rapid reinforcement learning. However, DEFT focuses on adapting within a fixed morphology (a dexterous five-fingered hand) and does not address cross-morphology generalization.

In contrast, RoboMorph uses large language models and evolutionary algorithms to optimize robot designs for tasks. While it generates novel morphologies, it assumes simulators for evaluation and does not enable policy reuse across different robots.

Closer to our goal, MetaMorph trains a universal Transformer-based controller conditioned on morphology tokens, achieving zero-shot transfer across modular robots. However, it relies on massive offline training and lacks online adaptability. Our work explores simpler, lightweight alternatives including morphology concatenation, graph-based embeddings, and task-space decomposition.

Other related methods in multi-task and multi-agent learning incorporate morphology or dynamics parameters into the policy input. Our CMV method follows this conditioning approach. Inspired by physical systems modeling, our GNN approach encodes the kinematic structure as a graph, though we find that such methods may require greater morphological diversity to generalize effectively.

In summary, while prior work has explored morphology optimization, universal control, or fine-tuning, our contribution lies in directly comparing several morphology-aware imitation learning strategies under a unified setup, evaluating their ability to generalize across real structural differences with moderate data.

3 Method

We aim to generalize imitation policies across robot morphologies. Our pipeline begins with training a base policy on a single robot, and extends to multi-robot training via various morphology-aware strategies. All policies are trained via Behavior Cloning (BC) using a neural network with a Mixture Density Network (MDN) output that predicts joint velocities.

Base Policy Architecture

Each demonstration consists of RGB images from a front and wrist-mounted camera, along with joint angles and end-effector pose. The action is a vector of joint velocities. We collected 100 expert demonstrations per task per robot.

The base policy $\pi_\theta(a_t \mid s_t)$ comprises dual streams: CNN encoders process visual inputs, and fully connected layers handle low-dimensional states. The fused feature vector is passed to an MDN head that models a mixture of Gaussians over joint velocities, capturing multi-modality in demonstrations. The model is trained via negative log-likelihood loss.

We first train the base policy on Panda robot data, using it in the Direct Transfer baseline. For unified policies, we train on all robots jointly while adjusting input structure to incorporate morphology.

Baselines

Per-Robot Policies (PRP): Each robot is trained separately on its own data, offering an upper-bound on task performance without generalization. This allows the policy to fully specialize to its morphology.

Direct Transfer (DT): We directly deploy a Panda-trained policy on the UR5e and iiwa without adaptation. Since UR5e has 6 DoF and the Panda policy expects 7, we truncate the output and zero-pad the UR5e's state input. This naive setup highlights the difficulty of ignoring morphology differences.

Figure 1: Task Overview.

Morphology-Aware Architectures

We explore four unified policy variants, all trained on the combined dataset from three robots. Each integrates morphology information differently:

1. Concatenated Morphology Vector (CMV): We construct a fixed-length descriptor m containing link lengths and DoF count. For robots with fewer joints (e.g., UR5e), the extra entries are padded with zeros. This vector is concatenated with the observation vector (either with state or after visual-state fusion) and passed to the network. The policy becomes $\pi(a_t \mid s_t, m)$, allowing the network to condition behavior on robot structure. However, zeros in m may be misinterpreted as valid inputs.

2. CMV with Mask (CMVM): To mitigate ambiguity, we add a binary mask b where $b_i = 1$ if m_i is real, 0 if padded. The input becomes $(s_t; m; b)$. This disambiguates padded versus meaningful morphology values, helping the policy avoid learning from phantom joints. Architecturally, CMVM mirrors CMV with a slightly longer input vector. In practice, the performance gain from masking was marginal.

3. GNN-Based Morphology Embedding (GNN-ME): Rather than manually encoding morphology, we use a Graph Neural Network (GNN) to learn an embedding from the robot's kinematic graph. Nodes represent joints/links, edges reflect connectivity. Node features include link lengths and joint types. The GNN outputs an embedding $\phi(m)$ via message passing and pooling. This embedding is concatenated with the observation input and fed into the MDN policy. We jointly train the GNN and the policy. However, with only three robot morphologies, the GNN struggled to generalize and likely overfit, showing poor transfer performance.

4. End-Effector Pose Policy + IK (EEPP): This method abstracts control to task space. The policy outputs $\Delta\mathbf{pose}$, a 6D end-effector displacement (position + orientation deltas), rather than joint velocities. A robot-specific inverse kinematics (IK) solver maps this into joint commands. The policy observes RGB images and low-dimensional states (excluding joint angles), and includes the current end-effector pose and target pose. The MDN output is translated via the IK solver for execution.

EEPP offers strong generalization because end-effector motion is largely robot-agnostic. It avoids morphology embedding altogether and leverages existing robot models to solve joint mappings analytically. However, this assumes the IK solver is available and accurate, and that tasks are defined in end-effector space. It may fail if the robot lacks sufficient dexterity to match the desired pose (e.g., UR5e failing to achieve 7-DoF wrist orientations). Despite these constraints, EEPP achieved the highest success in our experiments across all tasks and robots.

Summary

We propose and evaluate four morphology-aware policy architectures. CMV and CMVM provide explicit conditioning using robot parameters, while GNN-ME learns structure-aware embeddings. EEPP shifts to a morphology-agnostic action space, enabling high transferability. The unified training setup and MDN output enable all approaches to model multimodal expert behaviors. Our experiments demonstrate that lightweight morphology-aware conditioning improves generalization, but the best performance arises from factoring control into a morphology-invariant representation combined with analytical tools like IK.

4 Experimental Setup

We evaluate cross-robot generalization using three robot arms and three manipulation tasks in the RLBench simulation suite, which provides diverse tasks and a physics-based simulator built on CoppeliaSim.

Robots: We selected three robot models with varying morphologies:

- **Franka Emika Panda**: 7-DOF, torque-controlled, lightweight arm with wide motion range.
- **UR5e**: 6-DOF, position-controlled, lacking a spherical wrist joint, limiting its dexterity.
- **KUKA iiwa-14**: 7-DOF, torque-controlled industrial arm, with similar DoF to Panda but different link lengths and dynamics.

These robots span structural variations in DoF, scale, and control type.

Tasks: We chose three tasks of increasing complexity:

- **Reach Target**: Move the end-effector to a point in space—simple and contact-free.
- **Door Opening**: Grasp, rotate, and pull a door handle—requires multi-phase coordination.
- **Basketball in Hoop**: Pick up a ball and place it in a hoop—requires precision and height control.

Data Collection: For each task-robot pair, we collected 100 scripted expert demonstrations using RLBench's API, totaling 900 demos. Each demo includes synchronized RGB images (128×128) from front and wrist cameras, joint states, and joint velocities. State features and actions were normalized for training. We split data into 90% train and 10% validation sets.

Training: Policies were trained using Behavior Cloning with an MDN output, on a single GPU workstation using the Adam optimizer. Each trajectory contained 50–200 timesteps, yielding 150k data points across all robots and tasks. Unified policies were trained on the entire dataset; PRP and DT used only a single robot's data.

Evaluation: Each policy was evaluated in 20 randomized rollouts per task per robot. A trial is successful if the robot completes the task within a fixed time and meets the RLBench-defined success condition. For example, the basketball must land in the hoop, or the door must open past a threshold. For stochastic MDN outputs, we use the mean of the most probable Gaussian during evaluation. Success rates are averaged over 60 trials per robot (3 tasks × 20 trials), and we also report per-task results to assess task difficulty.

5 Results

We report success rates (fraction of successful rollouts) for each method across three manipulation tasks and three robot arms. Each task was evaluated with 20 randomized trials per robot. The full breakdown is shown in Tables 1, 2, and 3.

Table 1: Success rates on **Reach Target** task.

Robot	PRP	DT	CMV	CMVM	GNN-ME	EEPP
Panda	1.00	1.00	–	–	–	–
UR5e	1.00	0.00	0.10	0.20	0.05	0.95
iiwa	0.95	0.05	0.55	0.45	0.05	0.95

Table 2: Success rates on **Door Opening** task.

Robot	PRP	DT	CMV	CMVM	GNN-ME	EEPP
Panda	0.95	0.95	–	–	–	–
UR5e	0.90	0.00	0.00	0.05	0.00	0.95
iiwa	0.95	0.00	0.40	0.45	0.00	0.80

Table 3: Success rates on **Basketball in Hoop** task.

Robot	PRP	DT	CMV	CMVM	GNN-ME	EEPP
Panda	0.75	0.75	–	–	–	–
UR5e	0.60	0.00	0.00	0.00	0.00	0.65
iiwa	0.65	0.00	0.00	0.45	0.00	0.50

Qualitative Analysis and Discussion

Beyond quantitative success rates, we analyze how each method behaves and why.

Why Did Morphology Vectors Help?

The CMV and CMVM results show that conditioning the policy on a morphology vector m enables basic adaptation across robot structures. The policy likely learned distinct behavior modes based on m, adjusting outputs depending on robot-specific features (e.g., 7-DOF vs 6-DOF). This is analogous to multi-task learning where context variables modulate the policy. Without morphology input, the policy cannot distinguish why UR5e behaves differently, but with m, it learns this implicitly during training. Although CMV improved generalization, its performance remained below PRP, possibly due to network capacity limits or the challenge of encoding multiple optimal policies in one set of weights. More data or larger models might help, but full generalization may still fall short of PRP in high-complexity tasks.

Why Did the Mask Not Matter?

We hypothesized that a binary mask b would help disambiguate padded zeros in m. However, CMVM did not consistently outperform CMV. This suggests the network either inferred joint validity from the values in m or learned to down-weight the impact of padded entries. The mask might become more relevant with greater variation in morphology, but for our setup, its benefit was marginal. Thus, future implementations can omit the mask without significant loss, simplifying the architecture.

What Went Wrong with GNN-ME?

Despite its expressive power, GNN-ME performed poorly. With only three robot morphologies, the GNN likely memorized the graphs instead of learning generalizable embeddings. Since Panda and iiwa share the same topology and UR5e differs only in joint count, there was little structural diversity. The embeddings may have converged to near-constant vectors, offering no useful signal. Furthermore, the GNN architecture and node features (e.g., joint type, link length) may not have been optimal. Future work should explore pretraining the GNN or expanding the dataset with more diverse morphologies to fully leverage graph-based representations.

Why Was EEPP So Effective?

EEPP avoids direct morphology encoding by operating in task space. The policy outputs end-effector pose deltas, which are mapped to joint velocities using a robot-specific inverse kinematics (IK) solver. Since all robots share the same end-effector "language," EEPP generalizes naturally across morphologies. The IK solver handles the mapping to joint space, offloading complexity from the policy. Interestingly, EEPP matched or outperformed learned joint-space policies despite using a lower-dimensional output (6D pose vs 7D joint velocities), suggesting that this abstraction simplifies learning and improves sample efficiency. EEPP assumes access to accurate robot models and IK solvers, but for standard arms, this is a reasonable trade-off.

Task Difficulty and Performance Trends

Across methods, Reach Target was the easiest task, followed by Door Opening, with Basketball being the most difficult. Even PRP sometimes failed in Basketball due to minor placement errors. CMV-based policies struggled with precise control, especially in Basketball. EEPP, however, maintained high performance across tasks. Its ability to capture consistent end-effector strategies (e.g., position

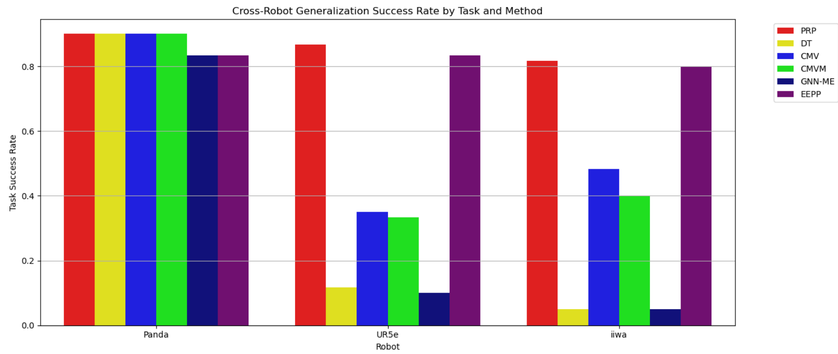

Figure 2: Visualization of results.

over hoop, then release) allowed for better generalization. In Door Opening, CMV and CMVM sometimes failed to pull the door after turning the handle, especially on UR5e, which lacks a wrist joint for fine orientation. These errors illustrate how morphology differences impact performance more significantly in complex, multi-phase tasks. This reinforces the need for morphology-aware policies or abstraction techniques like EEPP when deploying to new robots.

6 Discussion

Limitations

A limitation of our study is the relatively narrow set of robot morphologies: all three robots are serial manipulators with a maximum of 7-DOF. Thus, our conclusions may not generalize to significantly different systems, such as SCARA arms or humanoids. Our findings indicate that task-space policies like EEPP generalize well as long as the robot can physically reach the required end-effector poses. However, if a robot cannot execute the desired motion due to joint limits or reachability constraints, the policy would fail unless task strategies are adapted or certain tasks are excluded.

Another limitation is the simplified simulation environment. We did not consider dynamics or control delays, which are critical in real hardware. A kinematically sound end-effector policy might fail on a real robot with different response characteristics. Our dataset size was modest (100 demonstrations per task), and while sufficient for our analysis, some methods—particularly GNN-ME—might benefit from more data. We also did not perform thorough hyperparameter sweeps, meaning further tuning might improve results across all methods.

Implementation Insights

Developing and maintaining consistent pipelines across policy architectures required several engineering decisions. A key challenge was normalizing observations. Since joint ranges differ across robots (and UR5e has only 6 joints), we normalized joint angles to the range $[0, 1]$ based on each robot's configuration and used a padded dummy value for missing joints. Similarly, morphology descriptors like link lengths were normalized relative to a nominal scale to prevent input dominance from raw value magnitudes.

We also encountered instability in MDN training. In early stages, the network sometimes collapsed to a single mode. To address this, we initialized MDN output biases to encourage broad variance, which stabilized convergence. Although a deterministic regressor might suffice in some tasks, MDNs provided flexibility for modeling multimodal actions observed in demonstrations.

For EEPP, IK integration required fine-tuning. We used a resolved-rate control scheme where each policy output was translated into joint velocities via Jacobian-based IK. If IK failed due to singularities or large displacements, we applied step-size capping to ensure feasible motions. This worked well in our relatively unconstrained tasks, but future applications may require more robust solvers with damping or null-space objectives for success under tighter constraints.

7 Conclusion

This project explored morphology-aware imitation learning as a solution for generalizing manipulation policies across different robot embodiments. We developed and evaluated several methods that incorporate robot morphology into the policy network, ranging from simple feature concatenation (CMV), to structured representations (CMVM and GNN-ME), and a morphology-agnostic approach (EEPP) based on task-space control with robot-specific inverse kinematics.

Through experiments on three robot arms (Franka Panda, UR5e, KUKA iiwa) and three diverse tasks, we found that conditioning on morphology can significantly improve cross-robot generalization. The CMV approach demonstrated moderate success in transferring skills to unseen robots without retraining, confirming the value of explicit morphology inputs. However, GNN-ME underperformed in our setting, likely due to limited data and insufficient structural variation. This highlights the need for either larger morphology datasets or auxiliary pretraining when using learned embeddings.

The most effective strategy was to reformulate the control problem altogether. Our EEPP method—where the policy outputs end-effector pose changes and relies on an analytical IK solver for joint control—achieved near-specialist performance on all robots and tasks. This suggests that abstraction into a shared task-space representation, combined with known physical models, can yield robust generalization even in zero-shot scenarios.

These findings offer practical guidance. For practitioners, incorporating task-space control and leveraging existing robot models (e.g., via IK solvers) may be more effective than relying on purely learned mappings. For researchers, our results indicate the promise of combining structural priors with learning-based methods. Future directions include exploring meta-learning for fast morphology adaptation, scaling to more varied robots, and deploying these techniques on real hardware.

In summary, we achieved our goal of improving policy generalization across different robot morphologies. While we initially aimed to explore online adaptation, our focus on zero-shot generalization through morphology conditioning and task-space abstraction proved highly successful. This work demonstrates the feasibility of cross-robot imitation learning and lays the groundwork for making robot learning algorithms more general, data-efficient, and deployable across diverse platforms.

8 Team Contributions

Since this is an individual project, all tasks including literature review, environment setup, algorithm implementation, experiments, and final report writing was completed independently by myself.

Changes from Proposal Our original proposal aimed to explore online adaptation—fine-tuning imitation policies on new robots using limited reinforcement learning. However, we encountered challenges in applying RL to RLBench tasks, particularly balancing policy retention with exploration in high-dimensional spaces. Given time constraints, we pivoted to comparing offline morphology-aware methods using behavior cloning, which still addresses cross-robot generalization with minimal new data. We also revised our task set to Reach, Door Opening, and Basketball for their reliable scripted demonstrations and increasing difficulty. Additionally, we dropped the Kinova Gen3 robot from our experiments due to integration challenges and redundancy with other 7-DOF arms. These adjustments enabled us to conduct a focused, meaningful study. In hindsight, emphasizing offline generalization proved effective and yielded valuable insights into morphology-aware learning. Incorporating online adaptation remains a promising direction for future work, but was beyond the scope of this project.

References

[1] A. Kannan, K. Shaw, S. Bahl, P. Mannam, and D. Pathak, "DEFT: Dexterous Fine-Tuning for Hand Policies," in *Conference on Robot Learning (CoRL)*, 2023. Available: `https://dexterous-finetuning.github.io`

[2] K. Qiu, W. Pałucki, K. Ciebiera, P. Fijałkowski, M. Cygan, and Ł. Kuciński, "RoboMorph: Evolving Robot Morphology using Large Language Models," in *ICLR 2025 Workshop on Foundation Models in the Wild*, 2025.

[3] A. Gupta, L. Fan, S. Ganguli, and L. Fei-Fei, “MetaMorph: Learning Universal Controllers with Transformers,” in *International Conference on Learning Representations (ICLR)*, 2022. Available: `https://metamorph-iclr.github.io/site/`

[4] M. Dong and J. Zhang, “A review of robotic grasp detection technology,” *Robotica*, vol. 41, pp. 3846–3885, 2023. doi: 10.1017/S0263574723001285.

[5] Y. Cong, R. Chen, B. Ma, H. Liu, D. Hou, and C. Yang, “A Comprehensive Study of 3-D Vision-Based Robot Manipulation,” *IEEE Transactions on Cybernetics*, vol. 53, no. 3, pp. 1682–1706, Mar. 2023. doi: 10.1109/TCYB.2021.3108165.