Bridging the Sim-to-Real Gap in Humanoid Dynamics via Learned Nonlinear Operators

Anonymous Author(s)
Affiliation
Address
email

Abstract

A central challenge in humanoid robotic control is bridging the gap between simulated and real-world physics to enhance robot learning. This gap often leads to execution failures due to discrepancies in dynamics, actuator behavior, and unmodeled perturbations. This gap manifests as nonlinear variations at the level of each individual motor, and becomes even more unpredictable when the humanoid interacts with objects of varying weights. To better align simulation and real-world physics for robot learning, we curate **SimLifter**, the first dataset specifically designed to bridge the sim-to-real gap, even under the increased variability introduced by diverse payload interactions. **SimLifter** contains 257,493 frames, and systematically captures multimodal signals—including joint positions, velocities, torques, motor temperatures, and equivalent torques—across four standard payloads, three humanoid robots, and two actuation frequencies, covering both isolated joint movements and full upper-body coordination. We further introduce **GAPONet**, a novel reinforcement learning framework designed to enable robust policy transfer from simulation to the real world. Based on Unstacked Deep Operator Network (DeepONet) and Reinforcement Learning (RL), **GAPONet** estimates the discrepancies between simulation and real-world executions of the same policy by learning a nonlinear operator over multiple force modalities. It also demonstrates strong extrapolative generalization to unseen robots and zero-shot actions. On previously unseen humanoid robots, **GAPONet** shows substantial gains in zero-shot motion tracking, improving accuracy across all joints by over 40% on average compared to PD control with default hyperparameters, and up to 50% under maximum payload. In the goal-directed object delivery task, **GAPONet** improves accuracy by 70% under maximum payload over direct target action execution.

1 Introduction

Policies trained in simulation benefit from GPU acceleration and parallelized sampling [11], allowing fast and scalable training in environments with approximated physical properties such as mass, friction, and damping. However, in real-world interactions, forces such as friction and inertia often diverge significantly from these idealized assumptions [18, 24]. Various mismatches give rise to the well-known sim-to-real gap—a discrepancy between simulated and actual physical behavior—which becomes increasingly pronounced when humanoid robots interact with objects of varying mass. Crucially, this gap grows in a nonlinear and unpredictable manner, posing a serious challenge for robust policy transfer [23]. Bridging this gap is therefore essential for advancing humanoid robot control and enabling reliable real-world deployment.

To improve transferability, many current approaches begin with system identification to tune simulation parameters [8, 2, 15], apply domain randomization during training [14, 18, 3], or inject noise into observation signals [7, 22, 13] (e.g., mass, velocity) to prevent policies from overfitting

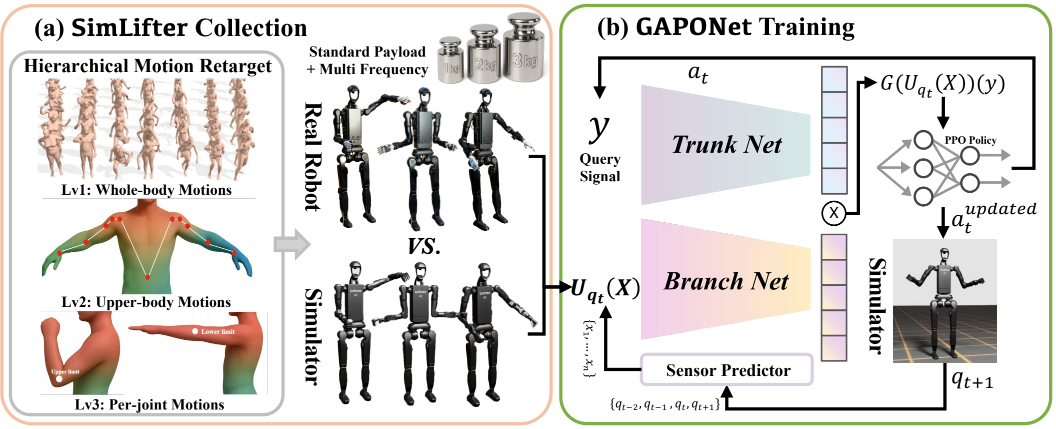

Figure 1: The overall structure of **SimLifter** and **GAPONet**. (a) shows the data collection process of **SimLifter**, where hierarchical motions are executed in both simulation and real-world platforms with standardized payloads to capture multi-modal paired data. (b) An overview of **GAPONet**, where the Branch Net and Trunk Net encode the robot's state and action, integrated with PPO for iterative learning.

to narrow distributions. Some also incorporate curriculum learning [10, 19] or progressively harder terrains to enhance policy robustness over time [16, 6]. While effective to some extent, these techniques require considerable manual tuning and domain expertise, making them labor-intensive and difficult to scale—especially for complex systems like humanoids.

A parallel line of work attempts to model real-world physics by learning dynamics from real-world data, using either observed states [17, 21] or actions [5]. While this approach holds promise, it typically requires large volumes of paired simulator-real data and struggles to model discrepancies for unseen actions. More importantly, these learned dynamics often fail to generalize across payloads of varying mass, limiting their ability to address the sim-to-real gap in diverse interaction scenarios. To overcome these limitations, we curate **SimLifter**, the first dataset specifically designed to bridge the sim-to-real gap, even under the increased variability introduced by diverse payload interactions. Different from prior datasets [20, 12, 1], **SimLifter** provides a structured and comprehensive collection of multimodal signals from humanoid upper-body interactions across varying payloads, actuation frequencies, and robot platforms, supporting detailed analysis and robust sim-to-real evaluation.

Building on our dataset, we propose **GAPONet**, which leverages a nonlinear operator [9] to model the sim-to-real gap in humanoid joint dynamics. This approach enables extrapolative generalization across varying payloads, mitigating overfitting to the collected data. **GAPONet** begins with a Sensor Predictor that encodes current state–action changes into distributions. These are then mapped into a latent space via the Branch Net, while the Trunk Net encodes two key query inputs — the payload and the current action. The resulting features are fused into a unified latent representation, from which the model predicts a delta action to compensate for the complex forces caused by discrepancies between simulation and real-world physics. We evaluate **GAPONet** on two tasks: (1) zero-shot motion tracking and (2) goal-directed object delivery, to assess its generalization and modeling capabilities on previously unseen humanoid robots. In zero-shot motion tracking, **GAPONet** improves accuracy across all joints by over 40% on average compared to sending target actions via API, and up to 50% under maximum payload. In the goal-directed object delivery task, it achieves a 70% accuracy improvement under maximum payload over direct target action execution. This paper makes three primary contributions:

- We curate **SimLifter**, the first dataset specifically designed to bridge the sim-to-real gap under the challenging variability introduced by diverse payload interactions. The data is collected across four distinct payloads and hierarchically structured into whole-body, upper-body, and per-joint levels.
- We introduce **GAPONet**, which models the sim-to-real gap in humanoid joint dynamics through a nonlinear operator, and achieves extrapolative generalization across different payloads, preventing overfitting to the collected data.

- We propose a **Sensor Predictor module** that replaces the fixed sensor input in traditional operator learning with a learnable sensor distribution, enabling parallel training within RL environments. This significantly reduces computational cost and makes the operator learning framework practically feasible.

2 `SimLifter` Dataset

`SimLifter` is the first dataset explicitly designed to bridge the sim-to-real gap under the variability introduced by diverse payload interactions. It records humanoid upper-body dynamics hierarchically — from individual joints to full upper-body motions — using three Unitree H1-2 robots with standardized end-effector weights (0–3 kg) and actuation frequencies of 50 Hz and 100 Hz. The dataset covers ten joints (six shoulders, two elbows, two wrists), with each sequence repeated three times across robots to reduce noise. In total, it contains 28.46 hours, 11,198 sequences, and 257,493 synchronized frames of joint positions, velocities, accelerations, torques, and motor temperatures. Each sequence is paired with a high-fidelity simulation replica, forming synchronized paired data for frame-level comparison of real and simulated executions. The distribution is shown in Fig. 2. Our experiments are further validated on a fourth H1-2 unit unseen during data collection.

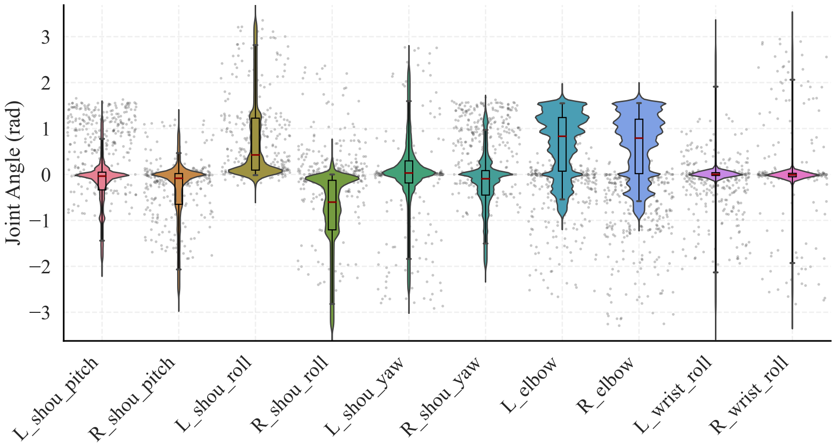

Figure 2: **Joint motion ranges.** The RainCloud plot illustrates the motion range of each upper-body joint by combining random samples (gray), box plots with red medians, and violin plots to reveal overall trends. The distributions span the valid ranges, and left–right joint pairs collectively approximate the electronic limits.

3 `GAPONet`

DeepONet [9]'s branch network encodes samples of the input function (e.g., joint trajectories), while the trunk network provides a query signal (e.g., future state or time) that enables effective generalization in a finite-dimensional space. In robotics, this framework can be integrated with RL algorithms to learn operator mappings between state-action trajectories and future dynamics, going beyond pointwise modeling of joint positions and velocities. Inspired by dynamic modeling [5], **`GAPONet`** predicts the delta action of each humanoid joint to compensate complex forces between the simulator and real-world physics [4]. Built on DeepONet and RL, it first employs a Sensor Predictor to encode current state–action changes into a set of distributions.

$$\mathcal{X} = f(A, Q, V, P, J, Q_{\text{real}}, V_{\text{real}}), \tag{1}$$

where A denotes the robot's target position, Q and V are the simulated DoF positions and velocities, and Q_{real} and V_{real} are the corresponding real-world values. The variable P indicates the payload during execution, while J represents the joint index in the humanoid's kinematic structure.

These inputs are mapped into a latent representation by the Branch Net:

$$\mathcal{B}(U_q(x)) = [\mathcal{B}_1(U_q(x)), \ldots, \mathcal{B}_p(U_q(x))], \tag{2}$$

where $x \in \mathcal{X}$ is the predicted sensor distribution, and $U_q(x)$ denotes the simulated state at the next step. Each component $\mathcal{B}_i$ encodes a distinct latent feature of the actuation state, enabling the network to decompose complex dynamics into interpretable subcomponents.

The Trunk Net encodes query signals consisting of the payload and current action,

$$Y = \{P, A\}, \quad \mathcal{T}(y) = [\mathcal{T}_1(y), \ldots, \mathcal{T}_p(y)]^\top, \tag{3}$$

which condition the latent space and align the actuator dynamics from the Branch Net with the target motion objectives.

The fused representation is obtained by combining Branch and Trunk features:

$$\mathcal{G}(U_q(x))(y) = \sum_{i=1}^{p} \mathcal{B}_i(x) \cdot \mathcal{T}_i(y), \tag{4}$$

yielding the delta action Δa^j for each joint j. This correction is added to the simulator's nominal command to bridge the sim-to-real gap:

$$s^{t+1} = f^{\text{sim}}\left(s^t_{\text{sim}},\, a^t_{\text{sim}} + \mathcal{G}(U_q(x^t))(a^t)\right). \tag{5}$$

4 Experiments

4.1 Zero-shot Motion Tracking

We validate our approach on ten motion sequences not included in the training set. The evaluation metric is the discrepancy in DoF positions between the simulator and the real robot, denoted as Q_{dis}. Specifically, Q_{dis} is defined as the integral of the difference between the target trajectory and the real robot's trajectory, where a smaller value indicates stronger gap-bridging capability. As shown in Fig. 3, we report Q_{dis} for eight upper-body joints under four payload conditions. The blue curve shows the natural mismatch between the simulator and the real robot without correction. The orange curve shows the results of [5], originally implemented with an MLP-based architecture (referred to as *MLP*). The green curve corresponds to the results of our proposed **GAPONet**.

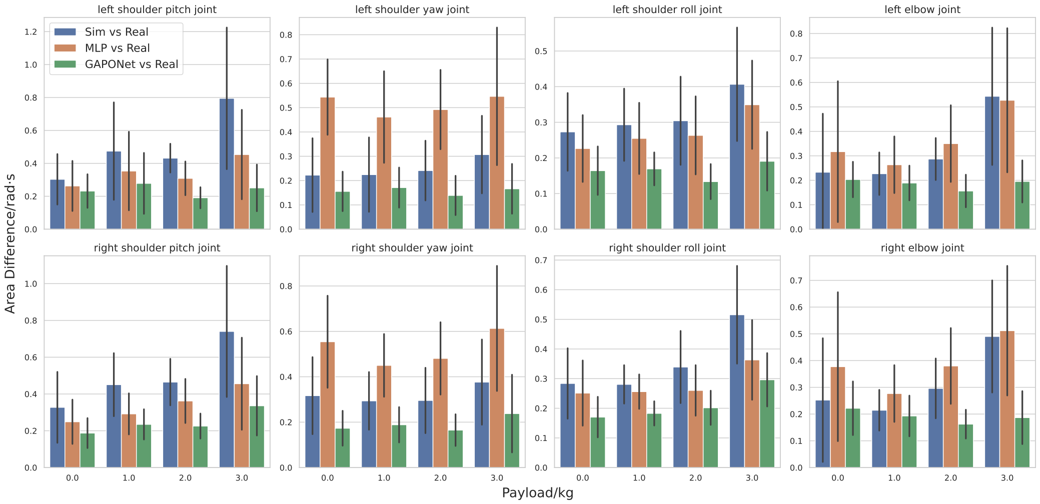

Figure 3: **Sim-to-real alignment (Q_{dis}) comparison across eight upper-body joints under four payload conditions. GAPONet** (green) achieves better trajectory alignment than MLP (orange) and the raw simulator output (blue).

We observe that the gap becomes more pronounced as the payload increases, while **GAPONet** achieves up to a 50% reduction in Q_{dis}. Compared to the MLP baseline, **GAPONet** also demonstrates stronger generalization, yielding better tracking accuracy even on zero-shot motion sequences.

4.2 Goal-directed Object Delivery

This task requires the robot to reach a specified global position while carrying payloads of varying mass. The challenge is to maintain accuracy under heavy loads, where gravitational effects make stable positioning difficult. As shown in Tab. 1, we evaluate four payload conditions across two models. Compared to the simulator's direct outputs, our approach significantly reduces the sim-to-real gap, enabling the robot to stably maintain the commanded position. **GAPONet** outperforms the MLP baseline with over 40% error reduction along the X and Y axes, and a **70% reduction along the Z axis**, effectively mitigating gravitational impact. These results demonstrate **GAPONet**'s robustness in bridging the sim-to-real gap under heavy payloads.

Table 1: Performance comparison of Dynamic Lifting and Lowering under varying payloads. The table presents positional deviations (X, Y, Z) for both **GAPONet** and MLP-based baselines and compare them with the direct output of simulator.

	with GAPONet			**with MLP**			**simulator**		
Payload	**X-offset (cm)**	**Y-offset (cm)**	**Z-offset (cm)**	**X-offset (cm)**	**Y-offset (cm)**	**Z-offset (cm)**	**X-offset (cm)**	**Y-offset (cm)**	**Z-offset (cm)**
0 kg	**0.3564**	0.6274	**0.4797**	0.4298	**0.6097**	1.7068	0.6733	0.6450	1.9698
1 kg	**0.3506**	0.7331	**0.4885**	0.4581	**0.5801**	1.7142	0.5424	0.7554	1.9978
2 kg	**0.3725**	**0.7357**	**0.5759**	0.4363	0.7374	1.8323	0.8149	1.2051	2.1376
3 kg	**0.4597**	0.7855	**0.6214**	0.4726	**0.7113**	2.1137	1.1455	1.2520	1.9269

4.3 Ablation

To better understand the contribution of different design choices in **GAPONet**, we conduct extensive ablation studies across five key dimensions Tab. 2: input structure, control frequency, observation history length, delta action duration, and prediction formulation. For fair comparison, all models are trained on **SimLifter** and evaluated using the same metric—mean per-joint angular error (MPJAE)—under different payload conditions.

Table 2: Ablation study results across input structure, frequency, model design, and prediction targets. **GAPONet** consistently outperforms other variants.

Ablation Setting	**Obs Dim**	**0kg MPJAE (rad) ↓**	**1kg MPJAE (rad) ↓**	**2kg MPJAE (rad) ↓**	**3kg MPJAE (rad) ↓**
(a) Payload Ablation					
GAPONet w p	$Obs \in R^{141}$	0.016	0.017	0.018	0.019
GAPONet w/o p	$Obs \in R^{140}$	0.013	0.012	0.013	0.015
(b) Frequency Ablation					
GAPONet @ 100Hz	$Obs \in R^{140}$	0.019	0.019	0.023	0.021
GAPONet @ 50Hz (Ours)	$Obs \in R^{140}$	0.013	0.012	0.013	0.015
(c) History Length Ablation					
History Length = 2	$Obs \in R^{80}$	0.016	0.016	0.017	0.019
History Length = 6	$Obs \in R^{200}$	0.015	0.016	0.016	0.019
History Length = 8	$Obs \in R^{260}$	0.017	0.017	0.018	0.021
History Length = 4 (Ours)	$Obs \in R^{140}$	0.013	0.012	0.013	0.015
(d) Action Duration Ablation					
Action Duration = 3	$Obs \in R^{140}$	0.017	0.016	0.017	0.018
Action Duration = 5	$Obs \in R^{140}$	0.015	0.016	0.017	0.019
Action Duration = 1 (Ours)	$Obs \in R^{140}$	0.013	0.012	0.013	0.015
(d) Prediction Ablation					
Predict S^{t+1}	$Obs \in R^{140}$	0.016	0.017	0.018	0.020
Predict $S^{t+1} - S^{t}$ (Ours)	$Obs \in R^{140}$	0.013	0.012	0.013	0.015

5 Conclusion

This work tackles the sim-to-real gap in humanoid control by introducing **SimLifter** and proposing **GAPONet**. **SimLifter** systematically captures multimodal joint-level signals across varying payloads, frequencies, and humanoid platforms, providing the first structured benchmark for analyzing and mitigating actuator-level discrepancies. Building on this foundation, **GAPONet** leverages operator learning with DeepONet to estimate and correct nonlinear mismatches between simulation and real executions, enabling robust policy transfer. Experiments show that **GAPONet** not only achieves over 40% improvements in zero-shot motion tracking but also delivers up to 70% accuracy gains in goal-directed object delivery tasks, demonstrating its generalization ability and effectiveness in bridging the sim-to-real gap. Additional details on model design, training procedures, and extended datasets with new experiments will be released in subsequent versions of this work.

References

[1] AgiBot-World-Contributors, Bu, Q., Cai, J., Chen, L., Cui, X., Ding, Y., Feng, S., Gao, S., He, X., Hu, X., Huang, X., Jiang, S., Jiang, Y., Jing, C., Li, H., Li, J., Liu, C., Liu, Y., Lu, Y., Luo, J., Luo, P., Mu, Y., Niu, Y., Pan, Y., Pang, J., Qiao, Y., Ren, G., Ruan, C., Shan, J., Shen, Y., Shi, C., Shi, M., Shi, M., Sima, C., Song, J., Wang, H., Wang, W., Wei, D., Xie, C., Xu, G., Yan, J., Yang, C., Yang, L., Yang, S., Yao, M., Zeng, J., Zhang, C., Zhang, Q., Zhao, B., Zhao, C., Zhao, J., and Zhu, J. (2025). Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*. 2

[2] Åström, K. J. and Eykhoff, P. (1971). System identification—a survey. *Automatica*, 7(2):123–162. 1

[3] Chen, X., Hu, J., Jin, C., Li, L., and Wang, L. (2021). Understanding domain randomization for sim-to-real transfer. *arXiv preprint arXiv:2110.03239*. 1

[4] Craig, J. J. (2009). *Introduction to robotics: mechanics and control, 3/E*. Pearson Education India. 3

[5] He, T., Gao, J., Xiao, W., Zhang, Y., Wang, Z., Wang, J., Luo, Z., He, G., Sobanbab, N., Pan, C., et al. (2025). Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. *arXiv preprint arXiv:2502.01143*. 2, 3, 4

[6] Heess, N., Tb, D., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, S., et al. (2017). Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*. 2

[7] Laskey, M., Lee, J., Fox, R., Dragan, A., and Goldberg, K. (2017). Dart: Noise injection for robust imitation learning. In *Conference on robot learning*, pages 143–156. PMLR. 1

[8] Ljung, L. (1998). System identification. In *Signal analysis and prediction*, pages 163–173. Springer. 1

[9] Lu, L., Jin, P., and Karniadakis, G. E. (2019). Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*. 2, 3

[10] Luo, S., Kasaei, H., and Schomaker, L. (2020). Accelerating reinforcement learning for reaching using continuous curriculum learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE. 2

[11] Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., et al. (2021). Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*. 1

[12] Mao, J., Zhao, S., Song, S., Shi, T., Ye, J., Zhang, M., Geng, H., Malik, J., Guizilini, V., and Wang, Y. (2024). Learning from massive human videos for universal humanoid pose control. *arXiv preprint arXiv:2412.14172*. 2

[13] Matas, J., James, S., and Davison, A. J. (2018). Sim-to-real reinforcement learning for deformable object manipulation. In *Conference on Robot Learning*, pages 734–743. PMLR. 1

[14] Mehta, B., Diaz, M., Golemo, F., Pal, C. J., and Paull, L. (2020). Active domain randomization. In *Conference on Robot Learning*, pages 1162–1176. PMLR. 1

[15] Nelles, O. (2002). Nonlinear system identification. *Measurement Science and Technology*, 13(4):646–646. 1

[16] Peng, X. B., Coumans, E., Zhang, T., Lee, T.-W., Tan, J., and Levine, S. (2020). Learning agile robotic locomotion skills by imitating animals. *arXiv preprint arXiv:2004.00784*. 2

[17] Shi, G., Shi, X., O'Connell, M., Yu, R., Azizzadenesheli, K., Anandkumar, A., Yue, Y., and Chung, S.-J. (2019). Neural lander: Stable drone landing control using learned dynamics. In *2019 international conference on robotics and automation (icra)*, pages 9784–9790. IEEE. 2

[18] Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 1

[19] Wang, X., Chen, Y., and Zhu, W. (2021). A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):4555–4576. 2

[20] Wu, K., Hou, C., Liu, J., Che, Z., Ju, X., Yang, Z., Li, M., Zhao, Y., Xu, Z., Yang, G., et al. (2024). Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*. 2

[21] Xiao, W., Xue, H., Tao, T., Kalaria, D., Dolan, J. M., and Shi, G. (2024). Anycar to anywhere: Learning universal dynamics model for agile and adaptive mobility. *arXiv preprint arXiv:2409.15783*. 2

[22] Zhang, H., Chen, H., Xiao, C., Li, B., Liu, M., Boning, D., and Hsieh, C.-J. (2020). Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in neural information processing systems*, 33:21024–21037. 1

[23] Zhang, X., Wang, C., Sun, L., Wu, Z., Zhu, X., and Tomizuka, M. (2023). Efficient sim-to-real transfer of contact-rich manipulation skills with online admittance residual learning. In *Conference on Robot Learning*, pages 1621–1639. PMLR. 1

[24] Zhao, W., Queralta, J. P., and Westerlund, T. (2020). Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, pages 737–744. IEEE. 1