Proceedings of CONF-MLA 2025 Symposium: Intelligent Systems and Automation: AI Models, IoT, and Robotic Algorithms
DOI: 10.54254/2755-2721/2025.LD26106

A Survey of Domain Adaptation in Robotics Using Diffusion Models

Xinyu Huang

Faculty of Engineering, University of Sydney, Sydney, Australia
xniyuhimself@gmail.com

Abstract. The successful deployment of intelligent robotic systems in the real world is often hampered by the “sim-to-real” gap, the discrepancy between simulated training environments and the complexities of reality. This gap arises from imperfect modeling of physics, rendering artifacts, and sensor noise, leading to policies trained in simulation failing to generalize. Domain adaptation techniques aim to bridge this gap, and recently, diffusion models have emerged as a powerful new paradigm for this task. This survey provides a comprehensive overview and contextual analysis of the application of diffusion models for domain adaptation in robotics. The paper begins by introducing the fundamental concepts of the sim-to-real gap and tracing the evolution of adaptation techniques, from domain randomization and adversarial methods to the current state of the art. The paper then presents a literature survey of recent works, categorizing them by their application in key robotics domains. Following this, a focused and in-depth case study provides a detailed walk-through of specific, influential methods, situating them within the landscape of prior work to highlight their core innovations. This survey then delves into a multifaceted discussion of the current challenges and open problems, including the critical trade-offs between computational efficiency and real-time performance, the debate surrounding generalization versus memorization, and the paramount issues of safety and reliability. The survey concludes by summarizing the state of the art and offering a perspective on the future directions of this rapidly evolving field, which is fundamentally reshaping how the industry approaches robust robotic learning.

Keywords: Domain Adaption, Diffusion Model, Generative Adversarial Networks (GANs)

1. Introduction

The advent of deep learning has revolutionized the field of robotics, enabling machines to learn complex behaviors and perceive their environment with unprecedented accuracy. A common paradigm for training these robotic systems is to use simulation, which offers a safe, scalable, and cost-effective way to generate vast amounts of training data. In simulation, a robot can attempt a task millions of times without wear and tear, operate in hazardous environments without risk, and experience a diversity of scenarios that would be impossible to stage in the real world. This has been a key enabler for breakthroughs in areas like reinforcement learning. However, the promise of learning complex skills entirely within a digital twin is consistently challenged by the "sim-to-real gap."

This gap is not a single problem, but a collection of subtle and overt discrepancies that create a distributional shift between the training (source) domain and the deployment (target) domain.

The three major discrepancies are Visual Discrepancies, Physical (Dynamics) Discrepancies, Sensor Noise and Actuator Delay.

Historically, bridging this gap involved two main strategies. The first was to make the simulator more robust through techniques like Domain Randomization, where physical and visual parameters of the simulation are heavily varied during training, forcing the policy to learn domain-invariant features [1]. The second was to explicitly align the source and target domains, often using Domain-Adversarial training to learn features that are indistinguishable to a domain classifier [2], or using Generative Adversarial Networks (GANs) [3] to perform image-to-image translation, making simulated images look more "real" [4]. While foundational, these methods had significant limitations. Domain randomization can be a brute-force approach, sometimes creating unrealistic scenarios that don't effectively prepare the agent. GANs, though powerful, are notoriously unstable to train, often suffering from mode collapse, and can produce visually plausible but semantically inconsistent translations.

The significance of diffusion models in this context lies in a fundamental paradigm shift. Unlike GANs, which learn a direct, one-shot mapping from a noise vector to a data sample, diffusion models learn a much more structured and controllable generative process. They work by iteratively refining a sample from pure noise, guided at each step by a learned model of the data distribution. This iterative refinement process is inherently more stable to train than the adversarial min-max game of GANs. More importantly, it allows for a more expressive and powerful form of adaptation. Instead of just aligning abstract features or performing a holistic image translation, diffusion models can manipulate the full data structure, directly transforming a data sample—whether it is an image or a full robot trajectory—from the simulated domain to the real domain.

This survey aims to explore this paradigm shift. The paper will provide a comprehensive overview of the current state of the art, first by surveying the breadth of applications and then by conducting a focused analysis of key methods that exemplify this new approach. Through a detailed walkthrough of their methodologies, this survey will illuminate not only the advancements but also the specific, new challenges that arise from this powerful class of models, ultimately providing a deep, contextualized understanding of this frontier in robotics research.

2. Background: from adversarial methods to diffusion models

To appreciate the contribution of diffusion models, it's essential to understand the landscape of techniques they are improving upon.

2.1. Former techniques

As introduced by Tobin et al. [1], the core idea of DR is to expose a learning agent to such a wide variety of simulated conditions that the real world appears as just another variation. This is achieved by randomizing textures, lighting conditions, camera positions, and even the physics parameters like mass and friction. By training on this highly varied data, the network is forced to learn features that are robust to these changes and thus more likely to generalize to the real world. This is a powerful, brute-force method that requires no real-world data, but its success hinges on the assumption that the randomization range is wide enough to encapsulate the real-world domain. If not, a "reality gap" can emerge where the real world is an outlier, and the policy still fails.

Inspired by the success of GANs [3], domain adaptation techniques sought to explicitly align data distributions. CycleGAN [4] became a bench- mark for visual sim-to-real. It uses a pair of generators (G_A to translate from domain A to B, and G_B from B to A) and a pair of discriminators (D_A and D_B). A crucial element is the cycle-consistency loss: $x \approx G_B(G_A(x))$. This ensures that if an image is translated from sim to real and back to sim, it should look like the original, forcing the generator to preserve content rather than just producing an arbitrary realistic image. On the feature level, Domain-Adversarial Neural Networks (DANNs) [2] introduced a gradient reversal layer. In this architecture, a single feature extractor feeds into two heads: a task classifier (e.g., object recognition) and a domain classifier. During training, the gradient reversal layer reverses the gradient flowing back from the domain classifier's loss. This means the feature extractor is trained to minimize the task loss while maximizing the domain classifier's loss, effectively learning features that are good for the task but useless for telling the domains apart.

2.2. The rise of diffusion models

Diffusion models, first proposed in 2015 [5] and later refined into a highly practical form by Denoising Diffusion Probabilistic Models (DDPMs) [6], offer a different approach. They consist of two processes:

Forward process is a fixed Markov chain that gradually adds Gaussian noise to a data point x_0 over T timesteps. The step sizes are controlled by a variance schedule β_t . The distribution of x_t given x_0 can be computed in closed form, which is crucial for efficient training.

Reverse process is a learned neural network $p_\theta\left(x_{t-1}\middle|x_t\right)$ that aims to reverse this process. By applying Bayes' rule, this reverse transition can be shown to be a Gaussian as well, provided β_t is small. The model is trained to predict the mean and variance of this reverse transition.

The key insight of DDPMs [6] was a simplified training objective. Instead of predicting the full reverse transition, they showed that it's more effective to train the network (typically a U-Net) to predict the noise ϵ that was added to x_0 to get x_t The loss function becomes a simple mean squared error between the true noise and the predicted noise: $L_{simple} = E_{t,x_0,\epsilon}\left[||\epsilon - \epsilon_0\left(x_t, t\right)||^2\right]$. This made training remarkably stable and effective.

However, the need to perform hundreds or thousands of iterative steps made sampling prohibitively slow. This was a major barrier for robotics until the development of Denoising Diffusion Implicit Models (DDIMs) [7]. DDIMs formulated a more general, non-Markovian forward process. This allowed the reverse process to skip steps (e.g., generating x_{t-k} from x_t , enabling high-quality sample generation in as few as 10-50 steps—a 10-100x speedup that made diffusion models far more practical.

3. Literature survey

The application of diffusion models to robotics is a burgeoning field, with a growing body of work surveyed by Wolf et al. [8]. These applications can be broadly categorized into perception, manipulation, and data augmentation.

3.1. Perception

A primary use case is bridging the visual sim-to-real gap. For semantic segmentation, De Rijk et al. [9] use a pre-trained diffusion model to perform style trans- fer, translating the style of synthetic images to match real-world images while preserving the semantic content necessary for training a segmentation network. This is a significant advancement over prior GAN-based methods, which could sometimes distort object boundaries or semantic layouts during translation. By producing higher-fidelity and more semantically consistent training data, these diffusion-based techniques allow perception networks to learn more robust and generalizable features.

3.2. Manipulation

Diffusion models are also being used to generate robot behaviors. They are particularly well-suited for this because they can naturally represent multi-modal policies (i.e., tasks where there are multiple equally valid solutions). For grasping, Li et al. [10] proposed an adversarial layout-to-image diffusion model that can generate a diverse set of high-quality grasp poses for a given object. For trajectory planning, Chen et al. [11] frame cross-domain policy adaptation as a data pre-processing problem, using a diffusion model to transform trajectories from a source domain to match the properties of a target domain.

3.3. Data augmentation

A more general application is for data augmentation. Li & Tamar [12] introduce a method that allows for fine-grained control over the level of realism in generated images. By adding a “realism hyperparameter” to the diffusion process, their model can generate a continuous spectrum of images, from purely simulated to highly realistic, allowing for a more gradual and controlled adaptation to the target domain. This provides more flexibility than one-shot translation methods like CycleGAN.

4. In-depth case studies

This paper now turn to a deep analysis of two representative methods—DiffuBox and 3D Diffusion Policy—to provide a walk-through of their methodologies and contextualize their advancements.

4.1. Case study 1: DiffuBox - refining perception

The DiffuBox paper [13] addresses a critical and nuanced problem in 3D object detection for autonomous driving: domain shift in object geometry and sensor noise.

4.1.1. The challenge with former techniques

The landscape of 3D object detection from point clouds was built on foundational work like PointNet [14], which enabled deep learning directly on unstructured point sets. However, when applying these detectors across domains (e.g., training on the KITTI dataset from Germany and testing on the nuScenes dataset from Singapore/Boston), performance drops significantly. Prior methods for unsupervised domain adaptation, such as SF-UDA3D [15], often relied on complex, multi-stage training pipelines. For example, SF-UDA3D first trains a detector on the source domain. Then, it uses this detector to generate pseudo-labels for the target domain data. Finally, it retrains a “student” model on a combination of source and pseudo-labeled target data, using complex consistency losses between different augmentations of the target data. These methods could be brittle, require careful hyperparameter tuning, and treat adaptation as a monolithic retraining problem.

4.1.2. A walk-through of DiffuBox

DiffuBox [13] introduces a radically simpler, modular approach. It acts as a post-processing refiner for any existing 3D detector.

The DiffuBox methodology begins by taking a coarse 3D bounding box—defined by its 7 parameters (center, dimensions, yaw)—from a base detector, along with the LiDAR points in its immediate vicinity. Its key innovation is the immediate transformation of these points into a "normalized box view." This is achieved by setting the origin to the coarse box’s center and scaling the points relative to the box's dimensions, a crucial step that makes the refinement process domain-agnostic by design. By separating shape estimation from scale estimation, the model learns the canonical shape of an object relative to its bounding box, not its absolute size. This standardized representation then feeds into the core diffusion process, where a surprisingly simple Multi-Layer Perceptron (MLP), conditioned on the normalized point cloud (processed by a mini-PointNet) and a timestep embedding, iteratively denoises a noisy version of the 7 box parameters to produce a final, refined box.

The primary advancement of this approach is its function as a zero-shot refiner that can be trained once and applied to new domains without any retraining, significantly improving localization accuracy. This modularity provides a significant engineering advantage over complex, end-to-end retraining pipelines. However, this design has notable limitations. First, its performance is fundamentally capped by the upstream detector, as it cannot fix false negatives; if the base detector misses an object, DiffuBox has nothing to refine. Second, it adds computational overhead, as the iterative sampling process, even with DDIM-style speedups, introduces latency that is a critical concern for real-time systems.

4.2. Case study 2: 3D Diffusion Policy - adapting actions

The 3D Diffusion Policy (DP3) [16] is a landmark paper in applying diffusion models to robotic manipulation, tackling the problem of learning from a small number of demonstrations.

4.2.1. The challenge with former techniques

DP3 improves upon a long line of research in imitation learning. A key paradigm is Behavioral Cloning (BC), which learns a direct mapping from observations to actions but is plagued by compounding errors. Most BC methods relied on 2D images, making them highly sensitive to visual domain shifts. Behavioral Cloning from Observation (BCO) [17] attempts to learn without explicit action labels but is even more challenging.

4.2.2. A walkthrough of 3D Diffusion Policy

DP3 [16] makes several key design choices to overcome these challenges.

The most fundamental contribution of 3D Diffusion Policy is its use of 3D point clouds from a single depth camera as its visual input, a design choice that builds on the success of methods like PointNet [14] to provide inherent invariance to viewpoint and texture—two of the biggest problems in sim-to-real. This robust 3D observation, along with the current robot state, serves as the conditioning for the policy itself, which is a conditional diffusion model designed to generate a short, future horizon of robot actions using a 1D U-Net that operates over the temporal dimension of the action sequence. At inference time, this policy executes an iterative refinement process: it starts with a randomly generated, noisy action sequence and, over a set number of steps, repeatedly applies the denoising network, which uses the conditioning variables to produce a slightly less noisy version of the actions until a final, smooth, and executable sequence is generated.

This architecture's primary advancement is its remarkable data efficiency, achieving high success rates on complex tasks with as few as 10-40 demonstrations—a dramatic improvement over prior methods that often required hundreds. However, this breakthrough comes with significant limitations. The most immediate is the slow inference speed inherent to the iterative denoising process. More fundamentally, a recent critique by Stone et al. [18] has sparked a crucial debate, arguing that the success of diffusion policies in low-data regimes might be attributable to a form of "sophisticated memorization" rather than true generalization. The argument is that the model acts like a powerful nearest-neighbor search, finding the closest training observation and recalling its associated action sequence. This has profound implications for safety, as the policy might be brittle and unpredictable when encountering novel scenarios not well-represented in its limited set of demonstrations.

5. Discussion: the shifting paradigm and its challenges

The deep dives into DiffuBox and DP3 illuminate the paradigm shift introduced by diffusion models, but also highlight pressing challenges.

5.1. The real-time imperative: latency and computational hurdles

The most immediate and practical challenge is the computational cost of iterative sampling. While models like DDIM [7] offer speedups, there remains a fundamental tension between sample quality and system latency. For a robot in a dynamic world, a 200-ms pause to generate an action is often unacceptable. This has spurred research into consistency models, which distill the knowledge of a diffusion model into a single-step generator, and knowledge distillation techniques to create smaller, faster policies.

5.2. The crisis of confidence: safety, verification, and the generalization debate

The debate sparked by Stone et al. [18] is perhaps the most significant long-term challenge. If these powerful policies are acting as high-capacity lookup tables, how can we verify their behavior or guarantee safety? Their performance on out-of-distribution states becomes dangerously unpredictable. This calls for new methods of verification. Perhaps the diffusion process itself can be leveraged: the magnitude of the predicted noise at each step could serve as a proxy for model uncertainty. A policy could be designed to recognize when it is in a low-density region of its training distribution and “ask for help” or revert to a simple, safe fallback behavior.

5.3. Architectural philosophies: end-to-end vs. modular design

The contrast between DiffuBox and DP3 highlights a key design choice. DiffuBox is a modular, disentangled solution. DP3 is an end-to-end solution. Modularity is often better for systems engineering—it’s easier to debug and upgrade individual components. However, end-to-end learning can, in theory, discover more optimal solutions by co-adapting perception and control. The success of both suggests there is no one-size-fits-all answer.

5.4. The data challenge: quality over quantity

While diffusion policies are lauded for their data efficiency, this places an even greater emphasis on the quality of the demonstrations. If the policy is essentially memorizing, then the demonstrations must be high-quality, diverse, and cover the expected state space well. A few bad demonstrations could poison the "lookup table" and lead to consistent failures. This shifts the engineering challenge from collecting massive datasets to curating smaller, higher-quality ones.

6. Conclusion

This survey has provided a comprehensive analysis of the emerging role of diffusion models in solving the sim-to-real problem in robotics. By first tracing the lineage of domain adaptation techniques from domain randomization and adversarial networks, this survey established the context for the paradigm shift that diffusion models represent. Through a broad literature review and a focused, in-depth case study of key methods like DiffuBox and 3D Diffusion Policy, this paper has detailed the specific mechanisms, advancements, and inherent limitations of this new approach. This survey has highlighted how these models are being applied to both perception and manipulation, while also critically examining the foundational challenges they introduce.

However, this advancement is not a panacea. The very properties that make these models powerful also lead to critical issues of computational latency and raise new, fundamental questions about safety, reliability, and the nature of generalization. The path forward involves a multi-pronged attack. We need continued algorithmic innovation for faster sampling, new theoretical frameworks for verification, and new system architectures, such as hybrid models that combine the generative power of diffusion with the formal guarantees of classical control or the causal reasoning of symbolic AI. By tackling these challenges head-on, the robotics community can harness the full potential of diffusion models to create truly robust, adaptable, and intelligent systems.

References

[1] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world, " in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 23–30.

[2] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marc- hand, and V. Lempitsky, "Domain-adversarial training of neural networks, " Journal of Machine Learning Research, vol. 17, no. 1, pp. 2096–2030, 2016.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets, " in Advances in Neu- ral Information Processing Systems (NeurIPS), 2014, pp. 2672–2680.

[4] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks, " in IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2223–2232.

[5] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsuper- vised learning using nonequilibrium thermodynamics, " in International Conference on Machine Learning (ICML), 2015, pp. 2256–2265.

[6] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models, " in Ad- vances in Neural Information Processing Systems (NeurIPS), 2020.

[7] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models, " in Interna- tional Conference on Learning Representations (ICLR), 2020.

[8] T. Wolf, E. Schperberg, and O. Kroemer, “Diffusion models for robotic manipulation: A survey, ” arXiv preprint arXiv: 2504.08438, 2025.

[9] T. De Rijk, N. Kooper, and D. M. Gavrila, “Style transfer with diffusion models for synthetic-to-real domain adaptation, ” arXiv preprint arXiv: 2505.16360, 2025.

[10] C. Li, Z. Chen, and H. Su, “Adversarial layout-to-image diffusion model for control- lable grasp generation, ” arXiv preprint arXiv: 2403.11459, 2024.

[11] L. Chen, C. Lu, and G. Lee, “xted: Cross-domain trajectory editing using diffusion models, ” in International Conference on Learning Representations (ICLR), 2025.

[12] K. Li and A. Tamar, “Continuous simulation-to-real transfer with diffusion models, ” in IEEE International Conference on Robotics and Automation (ICRA), 2024.

[13] X. Chen, Z. Liu, K. Z. Luo et al., “Diffubox: Refining 3d object detection with point diffusion, ” in Advances in Neural Information Processing Systems (NeurIPS), 2024.

[14] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation, ” in Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 652–660.

[15] C. Saltori, Z. Zhong, M. R. Oswald, and F. Tombari, “Sf-uda3d: Source-free unsuper- vised domain adaptation for 3d object detection, ” in IEEE International Conference on Computer Vision (ICCV), 2021, pp. 172–181.

[16] Y. Ze, G. Zhang, K. Zhang et al., “3d diffusion policy, ” in Robotics: Science and Systems (RSS), 2024.

[17] F. Torabi, G. Warnell, and P. Stone, “Behavioral cloning from observation, ” in International Joint Conference on Artificial Intelligence (IJCAI), 2018, pp. 4950– 4957.

[18] A. Stone et al., “Demystifying diffusion policies: Action memorization and simple lookup table alternatives, ” arXiv preprint arXiv: 2505.05787, 2025.