

Robot Learning: A Tutorial

Francesco Capuano  ... Adil Zouitine  Pepijn Kooijmans  Thomas Wolf  Michel Aractingi 

 École Normale Supérieure Paris-Saclay,  Hugging Face

Abstract



Hugging Face

Recent advancements in foundation models have enabled unprecedented progress in robotics, particularly in the development of increasingly more capable generalist robotics models. Leveraging modern learning-based approaches, the robotics community produced the first general models for tasks including locomotion and manipulation, and many said results have been openly released to the open-source community. The premise of generalist models is to learn control strategies across tasks and different robot embodiments, a long-standing challenge in robot learning. This work aims at being a tutorial on the most common techniques used for modern robot learning, covering both the conceptual underpinnings of the most prominent approaches, and pairing them with reproducible implementations using `lerobot` (<https://github.com/huggingface/lerobot>), the full-stack open-source robotics library developed by Hugging Face. This tutorial is designed to serve as a self-contained, structured reference for robotics practitioners working with `lerobot`, as well as a practical guide for researchers in the field of robot learning.

1 Introduction

1.1 Motivation: The Interdisciplinary Nature of Robotics in the Machine Learning Era

Robotics in 2025 sits at the intersection of classical model-based control, machine perception, and large-scale machine learning. Foundational problems in locomotion, manipulation, and whole-body control demand reasoning across rigid-body dynamics, contact modeling, planning under uncertainty, and high-dimensional function approximation. At the same time, end-to-end learning has matured from proof-of-concept demonstrations to systems that benefit from internet-scale multimodal pretraining and robotics-specific fine-tuning, closing the gap between laboratory benchmarks and deployment in unstructured settings. This tutorial embraces that interdisciplinarity: it treats modern robot learning as a synthesis of classical priors and data-driven policies rather than a replacement of one by the other.

1.2 Scope and Contributions of This Work

This document serves two purposes. First, it provides a concise, academically rigorous overview of core concepts in classical robotics that remain essential for principled system design (kinematics, dynamics, planning, and control). Second, it surveys contemporary learning-based methods for robotic control, with an emphasis on (i) reinforcement learning (RL) and imitation learning (IL), (ii) single-task policy architectures such as transformer-based action chunking and diffusion policies, and (iii) multi-task vision–language–action (VLA) models. Throughout, we complement exposition with *reproducible* implementation guidance using the open-source `LeRobot` framework, to lower the barrier between concept and practice while maintaining scientific rigor.

1.3 Structure of the Report

Section 2 reviews classical robotics foundations and their limitations in contact-rich, high-DOF regimes. Section 3 formulates learning-based control via RL and IL, highlighting problem setups, algorithms, and known failure modes (e.g., reward misspecification, simulation gaps, and safety constraints). Section 4 details single-task policy families (transformer chunking, diffusion) and their practical training recipes. Section 5 surveys multi-task VLA models (e.g., RT-1/RT-2, OpenVLA, $\pi_0/\pi_{0.5}$, and SmolVLA), together with integration patterns and experimental evaluation protocols. Section 6, not covered here, discusses emerging directions (e.g., world models and post-training) beyond the present scope. TLDR: This section provides a possible answer to the question as to why traditional methods based

on motion planning and purely physics-based robot control should be updated in light of more modern learning-based approaches. Here, we argue about the superiority of learning-based approaches versus purely dynamics-based approaches in contexts where primary concerns include (1) generalization to tasks and embodiments (2) reducing dependency on human expertise (3) leveraging historical trends on the production of data.

2 Artificial motion

Robotics is concerned with producing artificial motion in the physical world in a way that is useful, reliable and safe. Thus robotics is at its very core an inherently multidisciplinary domain: producing motion requires interfacing various hardware and software components, each at various levels. Defining a concept of usefulness typically depends on a profound understanding of the applicative domain considered, whereas ensuring reliable and safe execution more so relies on estimating the inherent uncertainty in the robot’s actions, guaranteeing adaptiveness of control while optimizing for performance, and more. Knowledge of mechanical, electrical, and software engineering, as well as rigid-body mechanics and control theory have therefore been quintessential in robotics since the field first developed. Lately, Machine Learning (ML) has proven useful in complementing said disciplines, making usage of the robotics data currently becoming more and more available. As a direct consequence of its multi-disciplinary nature, robotics developed as a rather wide array of methods, all concerned with the main purpose of producing artificial motion in the physical world.

Methods to produce robotics motion range from traditional *explicit* models—leveraging precise descriptions of the mechanics of robots’ rigid bodies and their interactions with eventual obstacles in the robots environment—to fully *learning-based* systems, offloading modelling the robot mechanics and rather treating motion as a statistical pattern to learn given multiple sensorimotor readings (Bekris et al., 2024). Between these two extrema, a variety of methods exists, with many borrowing traits from the opposite extremum to tackle specific challenges. On the one hand, as learning-based systems can benefit from information relative to the physics of particularly static tasks, Temporal Difference (TD) methods have been complemented with Model-Predictive Control (MPC) (Hansen et al., 2022). On the other hand, explicit models may be relying on assumptions proving overly simplistic—or even unrealistic—in practice, as in the case of assuming perfect observability of a robot state using sensorimotor readings. In this context, feedback loops at the control level help mitigate the effects of poor state estimation, complementing the motion planning process with discrepancy-from-target information, similarly to how loss functions are used in statistical learning. Figure ?? graphically illustrates the most prominent models in general-purpose applications, and we refer the interested reader to Bekris et al. (2024) for a comprehensive overview of both general and application-specific methods for motion generation. Aiming at introducing the inherent benefit of learning-based approaches in the context of an increasing availability of robot-data—the main focus of this tutorial—in the rest of this section we will present an illustrative example for the task of manipulation using a rather simplistic technique in the broader context of traditional robotics: inverse kinematics. While far from more advanced techniques that have proven groundbreaking in specific real-world applications (among many others, (Hansen et al., 2022; Burrige et al., 1999)), we believe the basics primitive of a full description of movement provide sufficient intuition as to the motivation of learning-based methods in the context of modern robotics.

2.1 Different kinds of motion

At its core, robotics deals with producing motion via actuating joints connecting nearly entirely-rigid links. A key distinction between focus areas in robotics is based on whether the generated motion modifies (1) the relative state of the robot with respect to its environment, (2) the absolute state of the environment or (3) both relative and absolute state (Figure ??). For instance, (1) may consist in changes in the robot’s physical location within its environment. Generally, modifications to a robot’s location within its environment may be considered instances of the general *locomotion* problem, further specified as *wheeled* or *legged* locomotion based on whenever a robot makes use of wheels or leg(s) to move in the environment. Further, (2) are typically achieved *through* the robot, i.e. generating motion to cause a perform an action inducing a desirable modification, effectively manipulating (*manipulation*) the environment. Manipulation is a well studied problem class in robotics, nowadays solved in practice for static and repetitive manipulation tasks, typically performed in controlled and precisely studied scenarios for instance via industrial robots employed at various stages of manufacturing processes. Lastly, an increased level of dynamism in the robot-environment interactions can be obtained combining (1) and (2), thus designing systems capable to move within *and* interact with their environment, falling under the category of *whole-body control*, and is characterized by a typically much larger set of control variables compared to either locomotion or manipulation alone.

Work in producing robots capable of navigating a diverse set of terrains demonstrated the premise of both dynamics

and learning-based approaches for locomotion (Griffin et al., 2017; Ji et al., 2023; Lee et al., 2020; Margolis et al., 2022), and recent works on whole-body control indicated the premise of learning-based approaches to generate rich motion in complex robot including humanoids (Zhang et al., 2024; NVIDIA et al., 2025). Manipulation has also been widely studied, particularly considering its aforementioned relevance for many impactful applications ranging from high-risk applications for humans (Fujita et al., 2020; Alizadeh and Zhu, 2024; Fujita et al., 2020) to manufacturing (Sanneman et al., 2020). While explicit models relying on precise descriptions of the dynamics and knowledge of the robot-environment system have been the key to most industrial robots currently in use, recent works leveraging learning-based techniques for manipulation proved particularly promising. For instance, Zhao et al. (2023) accomplished highly dexterous manipulation tasks using much less performant hardware compared to industrial baselines, all avoiding tuning from robotics experts (Zhao et al., 2023) by exclusively learning to manipulate objects from robotics data.

2.2 A Toy Example: (Planar) Manipulation

Manipulating an object first requires reaching it first, to then perform some kind of action. Robot manipulators typically consist of a series of links and joints, all connected to an *end-effector*. Links and joints are considered responsible for generating motion, while the end effector is instead used to perform specific actions at the target location (e.g., picking/releasing objects via closing/opening a gripper, using a specialized tool like a screwdriver or a measurement device). As such, at its very core, manipulation entails moving to the location of an object.

If we consider the (toy) example presented in Figure ??, then we can analytically write the end-effector’s position as a function of the control applied Figure ??, Figure ??

2.2.1 Overcoming Estimation Error via Feedback Loops

Adding feedback loops to a (simplified) dynamical system

Tuning gains is not immediate and rather cumbersome

2.3 Limitations

Modelling is hard, fragile and not scalable

It does not benefit from the growing amount of data being produced

3 Learning-Based Approaches to Robotics

3.1 Reinforcement Learning (RL) for Robotic Control

3.1.1 Problem Formulation and Control Objectives

We pose control as a Markov decision process (MDP) with state space \mathcal{S} , action space \mathcal{A} , transition kernel P , reward r , and discount γ . A policy $\pi_\theta(a | s)$ maximizes $J(\theta) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$. Practical objectives include setpoint tracking, task success, energy regularization, and safety constraints (e.g., joint limits, collision avoidance).

3.1.2 Policy Optimization Methods in Robotics

Modern practice spans on-policy (e.g., TRPO/PPO) and off-policy (e.g., DDPG/TD3/SAC) actor-critic methods, as well as model-based RL that learns or exploits dynamics for sample efficiency. Off-policy approaches are favored for real robots due to data efficiency and the ability to incorporate replay. Auxiliary techniques—domain randomization, dynamics identification, and constraint handling—facilitate sim-to-real transfer and safe exploration.

3.1.3 Practical Implementation: Training RL Policies with LeRobot

LeRobot provides standardized dataset and model abstractions, and can be integrated with established RL backends. A typical workflow comprises: (i) environment and observation/action specification (simulated or real), (ii) policy class selection (e.g., SAC/PPO wrappers), (iii) logging and evaluation hooks, (iv) sim-to-real calibration (actuation limits, latency compensation), and (v) deployment with safety interlocks. The framework’s dataset utilities simplify offline RL pretraining or behavior cloning initialization before online fine-tuning.

3.1.4 Limitations of RL in Real-World Robotics: Simulators and Reward Design

Notwithstanding successes in quadruped locomotion and manipulation, practical bottlenecks persist: expensive data collection, reward misspecification and shaping burden, partial observability, and safety during exploration. High-fidelity simulation reduces real-world trials but introduces transfer gaps; reward engineering often requires domain expertise and iterative tuning. Stable, sample-efficient, and safety-aware RL remains an active research frontier.

3.2 Imitation Learning (IL) for Robotics

3.2.1 Leveraging Real-World Demonstrations

IL bypasses online reward optimization by learning from expert trajectories. In visuomotor settings, demonstrations pair high-dimensional observations (RGB, depth, proprioception) with action sequences. Policy learning reduces to supervised learning under covariate shift, necessitating strategies that mitigate compounding errors.

3.2.2 Reward-Free Training and Data-Centric Perspectives

Data-centric IL emphasizes broad, diverse, and real-world demonstration corpora to improve generalization. Pretraining on heterogeneous demonstrations and fine-tuning on task-specific data can produce robust policies without explicit rewards, especially for contact-rich manipulation.

3.2.3 A Taxonomy of IL Approaches

Behavior Cloning (BC) minimizes one-step imitation loss; interactive IL (e.g., DAgger) aggregates data under the learner’s state distribution; adversarial/apprenticeship methods recover reward surrogates; and diffusion- or flow-based action generators model multimodal action distributions. Interactive feedback and on-the-fly corrections further reduce covariate shift in deployment.

4 Single-Task Policy Architectures

4.1 Action Chunking with Transformers

4.1.1 Model Architecture and Training Objectives

Action Chunking with Transformers (ACT) models short horizons of low-level actions as fixed-length “chunks” conditioned on recent observations and proprioception. A transformer parameterizes a distribution over action sequences, enabling parallel prediction of temporally coherent control signals. Supervised training on teleoperated demonstrations minimizes sequence-level losses (e.g., mean-squared error on joint velocities/torques), optionally with scheduled sampling or consistency regularization to improve rollout stability.

4.1.2 Practical Implementation in LeRobot

A practical ACT pipeline comprises (i) synchronized demonstration collection (vision and robot states) with precise timestamping, (ii) dataset serialization into a standardized episodic format, (iii) chunked windowing of action targets, (iv) transformer training with early stopping on validation rollouts, and (v) deployment with action-rate limiting and safety monitors. Careful calibration of control frequency and chunk length is critical for fine manipulation.

4.2 Diffusion-Based Policy Models

4.2.1 Generative Modeling for Action Sequences

Diffusion policies treat action generation as conditional denoising: starting from noise in action space, a learned score model iteratively refines to produce feasible control sequences conditioned on observations (and optionally goals or language). This formulation naturally captures multimodality (multiple valid strategies), scales to higher-dimensional actions, and supports receding-horizon execution.

4.2.2 Practical Implementation

Training proceeds by corrupting ground-truth action sequences and fitting a time-indexed network to denoise under teacher forcing; inference uses a small number of reverse steps with horizon MPC-style rollouts. In practice, datasets benefit from action normalization, viewpoint augmentation, and careful tuning of denoising steps versus control latency. Within LeRobot, diffusion policy trainers can be paired with standardized dataset loaders and evaluation callbacks to track success rates and robustness under perturbations.

5 Multi-Task Policies: Vision–Language–Action (VLA) Models in Robotics

5.1 Overview of Major Architectures: RT-1, RT-2, OpenVLA, π_0 , $\pi_{0.5}$, SmolVLA

VLA models integrate perception, instruction following, and action generation within a unified network. RT-1 demonstrated large-scale, robot-collected datasets and transformer policies capable of executing hundreds of distinct manipulation tasks; RT-2 extended this idea by co-training with web-scale vision–language corpora and representing actions as discrete tokens, improving semantic generalization to novel objects and instructions. OpenVLA adopts an open 7B backbone that fuses pretrained vision encoders with a language model and trains on large, diverse robot demonstrations, reporting strong cross-embodiment performance and efficient fine-tuning. The π_0 family introduces flow-matching action heads atop a pretrained VLM to produce continuous, high-frequency motor commands for dexterous skills; $\pi_{0.5}$ augments co-training for broader open-world generalization. Complementing these, SmolVLA targets efficiency and accessibility with a compact ($\sim 4.5 \times 10^8$ parameters) architecture and an interleaved attention “action expert” for low-latency control.

5.3 Practical Implementation: Integrating VLAs with LeRobot

Integration typically follows three phases. *Pretraining*: initialize the VLM backbone (e.g., SigLIP/DINOv2 features) and attach an action head (autoregressive, diffusion, or flow-matching) trained on heterogeneous demonstrations curated in a standardized format. *Task adaptation*: fine-tune with instruction-augmented, robot-specific data using low-rank adaptation where possible; calibrate action scaling, control-rate chunking, and observation encodings. *Deployment*: quantize or distill for on-robot inference; implement asynchronous inference stacks to decouple perception and control loops; add safety fallbacks and confidence gating. LeRobot dataset and model utilities simplify all three phases by providing consistent IO, evaluation tools, and model hosting.

5.4 Experimental Evaluation

A principled evaluation protocol should measure (i) success rate across held-out tasks and objects, (ii) language grounding fidelity under paraphrases and compositional instructions, (iii) cross-embodiment transfer (train/test robot mismatch), and (iv) robustness to distribution shift (lighting, clutter, distractors). Where possible, report zero-shot and few-shot adaptation performance, ablate pretraining datasets and backbones, and include real-world trials with fixed seeds and video evidence. Public, diverse datasets and standardized success criteria are critical to comparable results.

References

- Mohammad Alizadeh and Zheng H. Zhu. A comprehensive survey of space robotic manipulators for on-orbit servicing. *Frontiers in Robotics and AI*, 11, October 2024. ISSN 2296-9144. doi: 10.3389/frobt.2024.1470950.
- Kostas E. Bekris, Joe Doerr, Patrick Meng, and Sumanth Tangirala. The State of Robot Motion Generation, October 2024.
- R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential Composition of Dynamically Dexterous Robot Behaviors. *The International Journal of Robotics Research*, 18(6):534–555, June 1999. ISSN 0278-3649, 1741-3176. doi: 10.1177/02783649922066385.
- Jun Fujita, Daisuke Soda, Chotaro Murata, and Hiroyuki Tshari. Development of Robots for Nuclear Power Plants and Their Application to New Fields. 57(4), 2020.
- Robert J. Griffin, Georg Wiedebach, Sylvain Bertrand, Alexander Leonessa, and Jerry Pratt. Walking Stabilization Using Step Timing and Location Adjustment on the Humanoid Robot, Atlas. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 667–673, September 2017. doi: 10.1109/IROS.2017.8202223.

- Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal Difference Learning for Model Predictive Control, July 2022.
- Yandong Ji, Gabriel B. Margolis, and Pulkit Agrawal. DribbleBot: Dynamic Legged Manipulation in the Wild, April 2023.
- Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning Quadrupedal Locomotion over Challenging Terrain. *Science Robotics*, 5(47):eabc5986, October 2020. ISSN 2470-9476. doi: 10.1126/scirobotics.abc5986.
- Gabriel B. Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid Locomotion via Reinforcement Learning, May 2022.
- NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An Open Foundation Model for Generalist Humanoid Robots, March 2025.
- Lindsay Sanneman, Christopher Fourie, and Julie A. Shah. The State of Industrial Robotics: Emerging Technologies, Challenges, and Key Research Directions, October 2020.
- Chong Zhang, Wenli Xiao, Tairan He, and Guanya Shi. WoCoCo: Learning Whole-Body Humanoid Control with Sequential Contacts, November 2024.
- Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware, April 2023.