

# Demonstrate Once, Execute on Many: Kinematic Intelligence for Cross-robot Skill Transfer

Sthithpragya Gupta<sup>1\*</sup><sup>†</sup>, Durgesh Haribhau Salunkhe<sup>1†</sup>, Aude Billard<sup>1</sup>

<sup>1</sup>Learning Algorithms and Systems Laboratory, École Polytechnique Fédérale de Lausanne, Lausanne & 1015, Switzerland

<sup>†</sup>These authors contributed equally to this work.

Teaching robots new skills should be as natural as showing rather than programming. Learning from Demonstration (LfD) moves toward this goal by allowing users to guide a robot or sketch a desired motion, enabling learning without writing a line of code. Yet most LfD methods remain tied to the robot they were trained on. Changes in morphology, different link lengths, joint orientations, or limits often break the learned behaviour, making retraining unavoidable. Here we introduce a framework that endows robots with kinematic intelligence: an internal understanding of their own joint limits, singularities, and connectivity. Instead of correcting for these constraints after learning, we embed them directly into the control policy from the outset. The approach takes one or multiple demonstrations, extracts a globally stable dynamical system, and produces behaviours that remain valid across robots with different kinematic structures. Our method is grounded in a comprehensive analytical classification of non-cuspidal 3-revolute (3R) arms, which form the building blocks of many industrial manipulators. This classification enables a joint-space policy that preserves user intent while automatically adapting to robot-specific constraints. We validate the framework on a diverse set of simulated and real robots, redundant and non-redundant, with varied link geometries and joint configurations, and show that the demonstrated

**skill can be executed safely, smoothly, and consistently on all of them. By showing how analytical kinematic properties can be leveraged for cross-robot skill transfer, our work moves LfD closer to scalable, intuitive robot teaching for non-experts, enabling safe and reliable deployment without retraining.**

## INTRODUCTION

As robots become more common in our daily lives, from homes and hospitals to warehouses and factories, the ability to teach them new skills quickly and safely is becoming increasingly important. Instead of relying solely on expert programming, a more natural and intuitive alternative is to let users show robots what to do. Learning from Demonstration (LfD) provides this convenience (1–3). It allows users, expert or not, to teach a robot by simply demonstrating the desired behaviour. Demonstrations can take many forms: physically guiding the robot’s joints (kinesthetic teaching) (4), using a remote control (teleoperation) (5), or even drawing a desired path or motion in the workspace (as shown in Figure 1). This removes the need for coding and lowers the barrier for human-robot interaction. But while LfD simplifies how we teach robots, there’s a critical limitation: most of today’s LfD systems are tied to the specific robot they were trained on. If a user upgrades to a new robot, perhaps one with longer arms or different motion capabilities, the same taught skill may no longer work. Ideally, transferring skills between robots should be as seamless as syncing your preferences and apps when switching to a new phone or laptop. Unfortunately, that’s not the case today. The problem becomes even more difficult as new robots offer greater motion articulation, more degrees of freedom, and more complex joint configurations, each of which changes how the robot moves and what motions are feasible.

To overcome this challenge, we argue that robots need more than just the ability to mimic task demonstrations. They need to be equipped with what we call kinematic intelligence. This means that the robot must internally encode and account for its own kinematic constraints, joint limits, and feasible movement paths. Rather than relying on extensive demonstrations that implicitly avoid failures, our approach explicitly incorporates this structural awareness into the robot’s control policy. Kinematic intelligence plays a key role in enabling robots to generalize behaviour across different body structures. This is important because the human body rarely resembles the mechanical

structure of robots. Even among robots, different models often have very different link lengths and joint orientations, resulting in different types of unstable configurations (singularities). Yet the task being demonstrated, such as wiping a surface, picking up an object, or following a curved path, remains the same. To ensure that a behaviour learned on one system can be reused by others, we would typically need to reprogram or retrain the task with full knowledge of each robot's constraints. Transfer learning is a promising solution where a task is transferred by reusing a previously learned behaviour. However, transfer learning also faces challenges related to explainability, abstraction of transfer, and transfer metrics (6). Kinematic intelligence bridges this gap, enabling a system trained on one robot (or even from human-guided input) to function safely and effectively on a different robot without needing expert intervention.

**Challenges set forth by kinematic constraints in transfer learning:** Previous work in LfD has made significant progress in learning smooth and repeatable behaviours from demonstrations, particularly in simple or well-controlled environments. Early methods focused on directly imitating motion trajectories in the workspace, allowing robots to reproduce tasks like reaching, drawing, or object manipulation (7, 8). These approaches worked well when the target robot had similar physical characteristics to the one used during training. Some probabilistic models offer flexibility but still require separate handling of kinematic feasibility (9, 10). Other techniques, though more integrated, depend on detailed tuning or assumptions about the robot's workspace (11, 12). However, when the robot changes, or when its physical structure leads to limitations near joint limits or singular configurations, the learned behaviour may become infeasible or unsafe. Some approaches attempted to solve this by abstracting the task into a higher-level representation that is independent of the robot's embodiment (13–16). This abstract version is then adapted for each robot individually. While this approach shows promise, it often requires multiple demonstrations, extensive parameter tuning, or manual safety corrections near the robot's constraints (17, 18). Other research has explored learning mappings from workspace goals to joint configurations, commonly known as inverse models. These mappings work well in predictable areas of the robot's motion space, but they tend to fail near boundaries where movement becomes unpredictable or unstable. As a result, many methods include additional safety filters or online corrections to patch these issues in real time (19). Existing approaches typically emphasize task feasibility and constraints, but often treat

robot-specific constraints like singularities and joint limits as post hoc considerations rather than integral components of the learning process. While useful, these strategies treat the robot’s motion constraints as a separate concern rather than as part of the learning process itself. However, due to a lack of global understanding of the robot’s constraints, using numerical tools in post hoc, can make the robot follow paths that may contain unstable configuration(s) or hit a motion limit risking task failure or even endangering human safety (20–24).

More recently, some data-driven methods aim to build shared task spaces or embeddings that allow robots to learn behaviours that can be reused across different platforms. These methods have demonstrated generalization capabilities but often rely on large datasets, robot-specific fine-tuning, or access to every target robot during training [8–10]. This makes them less practical in real-world settings where only a few demonstrations are often available, as robot teaching is a time-consuming endeavour for a non-expert user. Overall, a gap remains in unifying robot-specific kinematic characteristics with generalizable learning frameworks that can function reliably with minimal demonstrations.

Many of these recent approaches are built around seven-degree-of-freedom (7-DoF) anthropomorphic robots, which have become standard in collaborative and humanoid robotics. Their popularity stems from their structural resemblance to the human arm and their enhanced flexibility, offering redundancy that allows for more natural and adaptive motion in complex environments, by offering more degrees of freedom (seven) than strictly necessary (six). However, redundancy also poses a challenge for learning and reproducing behaviour. Unlike non-redundant robots, redundant robots admit infinitely many joint-space solutions for a given task, see Figure 1(Top row), making it difficult to learn and generalize violation-free behaviours consistently across different configurations.

**Kinematic analysis to facilitate transfer across robotic manipulators** To analyze the behaviour of 7-DoF non-cuspidal manipulators, it is useful to decompose them into smaller kinematic units, or subchains, that isolate the functional components of motion. Most modern 7-DoF arms employ a spherical wrist, composed of three intersecting rotational joints that control the end-effector orientation. Anthropomorphic manipulators, widely adopted in research and certain industrial

applications such as the Kuka LWR iiwa and Motoman SIA series, exhibit non-cuspidal kinematic architectures (25). For such robots, inverse kinematic solutions (IKS) are separated by singularity boundaries in joint space, leading to well-defined, predictable transitions between configurations. The spherical wrist not only decouples orientation singularities from the positioning subchain but is also non-cuspidal, exhibiting two inverse kinematic solutions separated by  $\sin(\theta_w) = 0$ , where  $\theta_w$  denotes the second wrist joint. Hence, the analytical framework proposed for non-cuspidal  $3R$  (three revolute joints) positioning chains can be directly applied to the wrist subchain, providing certified, singularity-free orientation control. The remaining four joints form the positional subchain, redundant for 3D positioning, as shown in Figure 1(second row).

To manage the redundancy-induced ambiguity, the  $4R$  positional chain can be treated as a family of  $3R$  non-cuspidal subchains. By setting the angle of a redundant joint in the  $4R$  subchain to a specific value, we effectively eliminate one degree of freedom, reducing the  $4R$  subchain to a non-redundant  $3R$  subchain that is just sufficient for spatial positioning. Thus, the behaviour of the  $4R$  chain can be studied by analyzing the properties of its constituent  $3R$  subchains, where the redundant joint serves as a parameter indexing different members of the family. In this way, the entire  $4R$  positional chain can be interpreted as a family of  $3R$  subchains, each corresponding to a different value of the redundant joint angle. This creates a parameterized representation, where the value of the redundant joint angle acts as a parameter that indexes a continuum of  $3R$  subchains embedded within the original  $4R$  subchain. While each of these  $3R$  subchains abide by the same joint limit constraints, they can exhibit different kinematic properties, such as singularities, number of IK solutions, and the feasible joint space regions.

This decomposition provides a systematic means to analyze the core positional capabilities of 7-DoF anthropomorphic manipulators through a family of non-cuspidal  $3R$  subchains. These  $3R$  units constitute the kinematic foundation of the arm and form the basis for our global, constraint-aware analysis. Since the wrist joint angles depend functionally on the configuration of the positional sub-chain and can be computed reactively without ambiguity, the positional feasibility results obtained from these subchains can be directly extended to full end-effector pose control.

**Kinematic-aware learning of control policies:** Following the classification proposed by Burdick (26), 3R manipulators can be grouped as generic, non-generic, or degenerate according to the nature of their singularities. Generic robots exhibit well-behaved singular surfaces whose topology remains stable under small variations of link parameters. Non-generic robots correspond to special geometries where these surfaces intersect or bifurcate, leading to multiple branches or self-intersections. Degenerate cases arise typically in manipulators with symmetric link lengths or right-angle joint arrangements. In such configurations, two joint axes may align or intersect, reducing the arm’s effective mobility and leading to indeterminate motions if not explicitly modeled. Recognizing these degeneracies is therefore crucial: when known, they can be avoided for safety or exploited to improve task dexterity. Our framework systematically incorporates this awareness, embedding kinematic intelligence across all categories of non-cuspidal robots, generic, non-generic, and degenerate, thus ensuring constraint-aware and predictable behavior regardless of design variations.

We design safe and generalizable control strategies that are not tailored to a specific robot. Safety refers here to control strategies that guarantee violation-free task execution by embedding certified redirection mechanisms near kinematic constraints. To this end, instead of trying to remove or ignore a robot’s physical constraints as done in many LfD approaches, we embed these constraints analytically into the policy architecture. By doing so, we ensure that the learned behaviour is automatically adapted to each robot’s structure in a safe and stable way, see Figure 1(3r and 4th rows). Specifically, we use a topological and differential classification of singularities to understand how different parts of the robot’s configuration space are separated by constraints (refer to Figure 2). This classification allows us to identify singularity-free zones, distinguish between reachable and unreachable regions, and joint-space control policies that avoid constraint violations.

With such kinematic-aware embedding of the learned policies, even a single demonstration is enough to create a behaviour that is safe, robust, and executable on many different robots. We demonstrate this framework across a range of robots, including both redundant and non-redundant arms, each with different kinematic structures and singularity patterns. This is achieved without any retraining or architecture-specific reprogramming.

## RESULTS

We demonstrate transfer of skills in simulation with a series of non-redundant  $3R$  robots. For redundant robots, we consider the KUKA IIWA LWR 7 and the Maira robot, both seven-degree-of-freedom commercially available robot arms. The Maira robot has a peculiarity in that it has a degeneracy, but the degenerate configurations remain outside the feasible joint space. We observe across all robot configurations that the system successfully acquires a generalised policy of the user behaviour, resulting in smooth, stable, and physically feasible robot execution (refer to Figure 3 and Figure 8). Even with varying robot structures and joint constraints, the embedded kinematic intelligence ensured that the task execution remained safe and accurate, demonstrating the framework’s ability to generalize a skill across different robot bodies even from a single user demonstration. We further demonstrate redundancy parametrisation and how it facilitates behaviour transfer to redundant robots (refer to Figure 4). An overview of the challenges, a brief motivation to kinematic analysis, and behaviour transfer is presented in Movie S1.

### Kinematic intelligence

Kinematic intelligence refers to the embedding of analytical kinematic properties into the control framework using deterministic, fixed-compute algorithms. This enables certified, constraint-compliant path planning and policy synthesis that generalizes across robot embodiments without retraining.

A central result of this work is a theoretically grounded classification of all non-cuspidal  $3R$  manipulators into six distinct categories, each defined by the global structure of their kinematic constraints. This classification forms the backbone of our kinematic intelligence framework, enabling the synthesis of robot-specific control strategies that are inherently safe and generalizable.

The classification arises from an algebraic and topological analysis of the robot’s Jacobian determinant. The factorization and root structure of this determinant are determined by the robot’s design parameters, such as link lengths and joint offsets, which define the location and shape of singular configurations in the joint space. These singularities, along with joint limits, partition the joint space into distinct feasible regions: referred to in the remainder of this paper as **aspects** (27).

Each category captures a specific class of kinematic behaviours, characterized by how singular-

ity curves fold, loop, intersect to segment the joint space. Crucially, the structure of each class also dictates a corresponding near-boundary control strategy. Rather than treating constraint handling as an afterthought, the categorization informs how corrective actions should be embedded directly into the control policy. Once a robot’s class is known, the corresponding strategy can be instantiated without further tuning or demonstration, yielding a scalable method for constraint-compliant behaviour transfer across robot types.

Figure 2 provides an overview of the topological structure of the singularities and joint limits in each class. This structure is not just descriptive, it is operational. It allows us to build controllers that adapt dynamically to robot-specific constraints while preserving the fidelity of the demonstrated task.

### ***Robot Categorization***

Through a detailed algebraic analysis of the Jacobian determinant  $\det \mathbf{J}(\mathbf{q})$  where  $\mathbf{q}$  represents the robot’s joint configuration, we establish a structured classification scheme for noncuspidal  $3R$  robots.

The expression of singularities for  $3R$  robots is only affected by the actuation of the second and third joints ( $q_2$  and  $q_3$  respectively). Consequently, the kinematic constraints imposed by singular configurations can be analysed in a reduced space spanned by the second and third joint’s angle values, hereby referred to as the  $q_2 - q_3$  slice of the joint space. The expression for  $\det \mathbf{J}(\mathbf{q})$  can be decomposed into upto three irreducible factors, whose zero sets define closed branches on the configuration torus  $\mathbb{T}^2$ . Each factor exhibits distinct winding numbers  $(n_1, n_2)$ , denoting how many times they wrap around the torus generators associated with joints  $q_2$  and  $q_3$ . Each individual encirclement by the factor is a unqiue branch of that factor. Building on this topological insight, we introduce a novel symbolic notation of the form  $(n_1, n_2)[n_3, n_4]$ , which describes both the global winding behavior and the differential structure, where  $n_3$  and  $n_4$  capture the number of horizontal and vertical turning points, respectively that can occur on a branch of the factor. This notation extends the existing homotopy classification (28, 29), which relied only on the winding numbers. While this classification was exhaustive, it remained insufficient for planning motions along constraints, as robots within the same category could exhibit distinct differential properties-necessitating case-by-case treatment during execution. In contrast, our proposed categorization

incorporates both topological and differential properties, ensuring that robots within the same class exhibit identical constraint structures and thus permit a unified path planning strategy near singularities.

Applying this framework across both generic and non-generic noncuspidal  $3R$  robots, we identify six canonical robot categories that comprehensively describe the global structure of the joint space. To arrive at this categorisation, we inspect the factor(s) of the expression of  $\det \mathbf{J}(\mathbf{q})$  for the given robot and generate a description of their accompanying branches. The six categories as presented in Figure 2(B) are: (I) non-loop, non-intersecting factors with ONLY horizontal turning points - only factors with branches of type  $(1, 0)[\geq 2, 0]$ ; (II) non-loop, non-intersecting branches with NO horizontal turning points - factors with branch types  $(0, 1)[0, *]$  where  $*$  can assume any non-negative value; (III) non-loop intersecting branches - two or more factors with non-loop type branches that intersect; (IV) non-loop, non-intersecting branches with non-zero vertical turning points - factors with branch type  $(1, 0)[2, > 0]$ ; (V) loop non-intersecting branches - factor with branch type  $(0, 0)[2, \geq 2]$ ; and (VI) intersecting branches with a loop - intersecting factors where one of the factors has branch that makes a loop. Each class reflects a distinct mode of constraint-induced partitioning in joint space and corresponds to qualitatively different boundary-following behaviours. This classification organizes the singularities across  $3R$  manipulators, and provides a predictive basis for designing constraint-aware control policies that generalize across robot geometries. Multiple non-cuspidal robots with varying kinematic properties with their respective DH parameters are presented in Movie S2.

When a user demonstrates a task, the behaviour is encoded as a task-level objective, which each robot then interprets and executes in a constraint-aware manner based on its own categorisation. This approach contrasts with traditional LfD methods that either rely on post-hoc corrections or require multiple demonstrations tailored to each robot. Instead, our framework generalizes from a single demonstration by leveraging an internally encoded model of joint-space structure that is both robot-specific and reusable across the entire category. For example, if a demonstrated behaviour would cause a new robot to approach a singular configuration or violate a joint limit, the system automatically adjusts the trajectory to follow the boundary of the feasible region until it can safely return to the nominal path. This ensures continuity and stability in the robot's motion while faithfully reproducing the demonstrated behaviour.

In essence, the structural classification functions as a map that each robot uses to safely interpret and execute a shared task. By integrating this into the learning process, we move beyond treating kinematic constraints as external concerns and instead make them an intrinsic part of behaviour generalization. A comprehensive exposition of the mathematical framework of the categorization, defining properties of each robot class, their topological implications in joint space, the formalization of kinematic constraints and accompanying algorithms , is presented in the Supplementary Methods section ‘Robot Categorisation’.

### **Near-constraint strategy**

A key outcome of this work is the establishment of a one-to-one correspondence between the robot’s kinematic category and the control strategy that governs robot behaviour along **aspect** boundaries. Each joint-space aspect is bounded on at least one side by a factor of  $\det \mathbf{J}(\mathbf{q})$ , and on the remaining sides (if any) by the joint limits. While traversal along joint limits is straightforward since they form a linear variety in joint space, motion along the factor boundary is non-trivial and varies significantly across categories.

The core insight is that the traversal along the factor boundary within an aspect dictates the constraint-aware motion strategy, and this traversal is uniquely determined by the kinematic category of the robot. Each category is characterised by a key boundary-aligned direction(s), vector(s) intrinsic to the structure of the singularity factor, that governs motion near the constraint (see Figure 2(C)). This direction is projected onto the tangent of the boundary to yield the feasible motion along it. Crucially, this approach remains robust even in the presence of turning points or curvature reversals along the singularity branch: the projected motion does not stall or violate constraints.

Robots from Categories I, II, IV, and V exhibit a single key boundary-aligned direction viz- horizontal (along  $q_2$  axis) in Category I, vertical (along  $q_3$  axis) in Category II, a horizontal direction with intermittent vertical switching in Category IV, and along a closed-loop in Category V (see Figure 2(C1) through (C6)). In contrast, Categories III and VI are defined by a pair of distinct boundary-aligned directions. These arise due to the presence of two distinct factors in  $\det \mathbf{J}(\mathbf{q})$ , corresponding to intersecting singularities. Each factor contributes its own characteristic direction for boundary traversal. This is notably different from Category IV, where only a single

factor exists, despite the presence of mixed motion, meaning that both horizontal and vertical behaviours emerge along the same singularity branch, rather than from multiple factors. Note that the occurrence of multiple factors does not necessitate multiple directions. Robots from Category I can have upto two factors, yet since the branch types of these factors is identical ( $(1, 0)[\geq 2, 0]$ ), the horizontal direction suffices. We refer to the structured control strategies incorporating these category-specific boundary-aligned directions for governing the robot behaviour near constraints as *track cycles*. When the nominal dynamics modelled from the user behaviour generates a trajectory **Q** that temporarily exits the current aspect, convergence to the goal necessitates eventual reentry. The track cycle defines a stable control policy that takes effect between the first exit and last reentry configurations. Within this segment, motion is redirected along the boundaries of the aspect. Overall, the track cycle encapsulates the correct sequence of transitions across boundary regions, driving the robot from any boundary configuration back to a safe configuration, once again consistent with the original trajectory **Q**. We adopt a hybrid control framework that overlays track cycle policy onto the nominal joint-space dynamics learned from demonstration. These corrective policies activate preemptively as the robot approaches predefined safety margins around singular regions or joint limits, while following the trajectory generated by the nominal dynamics. The readers are encouraged to refer to section titled "Behaviour near Kinematic Constraints" in Supplementary Methods for further details on factors, their properties and track cycles.

## Transfer Across Robots

To evaluate cross-robot behaviour transfer, we considered a scenario where a user demonstrates a task, such as drawing shapes or letters, [hitting an object, rearranging an object and throwing](#), and different robots attempt to reproduce the behaviour. These robots vary in mechanical design: some are non-redundant (with just enough degrees of freedom), while others are redundant (with more joints than necessary). From this, our system generates robot-specific joint-space control policies that preserve the demonstrated intent while ensuring safety. These results validate the core hypothesis: by embedding structural constraints directly into the learning framework, a behaviour demonstrated just once can be safely and reliably transferred to robots with different kinematics, all without needing expert tuning or additional data.

## Transfer to Non-redundant robots

To evaluate how a single demonstration can be extended across multiple non-redundant robots with different kinematic structures, we tested our framework on four  $3R$  robots - one each from Categories 1 through 4. For each robot, the goal was to generate a control policy that generalised user-demonstrated motion but also respected robot-specific constraints like singularities and joint limits. As shown in Figure 3, our method models joint-space behaviours aspect-wise, where each aspect is a region free of singularities, and incrementally builds a constraint-aware policy. The process is presented for the four aforementioned robots in Figure 3 (A) through Figure 3 (D) respectively. The figure outlines each step of the framework: from modelling the workspace behaviour (Figure 3 Step 1), learning aspect-specific joint-space behaviours (Figure 3 Step 2), identifying feasible transfer trajectories (Figure 3 Step 3), and workspace-to-joint space transfer and updating the policy based on most novel transfer trajectories (Figure 3 Step 4). Finally, Figure 3 Step 5 formulates the resulting workspace motion that is faithful to the original demonstration while staying entirely within safe, constraint-compliant regions. Across all tested robot categories, the learned joint-space policy remains free from constraint violations without requiring retraining, enabling robust and scalable behaviour transfer to diverse non-redundant robots from just a single demonstration.

## Transfer to Redundant robots

To assess transfer to redundant robots, we consider a 7 DoF anthropomorphic arm. The user demonstrated that the end-effector trajectory is used to generate a control policy for the position control of these redundant arms.

The redundant robots can be parameterized by a chosen angle ( $q_R$ ) to represent a family of non-redundant  $6R$  robots. The architecture of the  $6R$  robot depends on the *arm angle* or *self-motion angle*. The arm angle is not always easy to define and interpret (30), and different methods have been proposed earlier to parameterize the redundant motion of  $7R$  robots (31), such as the *shoulder-elbow-wrist* (SEW) angle. It was shown in (25) that  $\det \mathbf{J}(\mathbf{q})$  of the redundant robots can be obtained as a function of  $q_R$ , implying that the redundant robot can be treated as a non-redundant robot at any arbitrary value of  $q_R$ . In practice, we discretise a set of candidate  $q_R$  values, run the full classification and connectivity analysis for the induced  $3R$  subchain at each candidate, and select a *fixed*  $q_R$  that

realises the task within a single feasible aspect with joint limits and singularity violation. This  $q_R$  is then kept constant during execution, so that, from the viewpoint of the positional chain, the robot behaves like a non-redundant 3R manipulator and no discontinuities due to online  $q_R$  switching are introduced.

The Figure 4 outlines the process of redundancy parametrisation into five behavioural modes for brevity

. The redundant 4R positional subchain is identified for each robot in Figure 4, and parametrised using different values for  $q_R$  -  $0^\circ$ ,  $30^\circ$ ,  $45^\circ$ ,  $60^\circ$ , and  $90^\circ$  in Figure 4(A). This yields the five behavioural modes, each characterised by its equivalent 3R subchains. All of the 3R subchains for the robot belong to Category III. The branch types for  $q_R = 0$  are  $(1, 0)[\infty, 0]$ , and  $(1, 1)[0, *]$  while for the rest are  $(1, 0)[\infty, 0]$ , and  $(0, 1)[0, *]$ . Figure 4(B) presents the varying kinematic constraints across these behavioural modes in joint space. At each  $q_R$  value, multiple IK solutions for the starting position of the 3R subchain will exist. Some of these configurations are feasible for task completion, Figure 4(B) and produce workspace behaviour Figure 4(C). However, some of the other IK solutions (silver) may not be feasible. For  $q_R = 0^\circ$ , and  $60^\circ$ , feasible behaviours are not detected as the goal joint configuration in the respective

aspect of joint space lies in the joint limit constraint boundary and hence safe task execution is not feasible. Notice that the feasible workspace Figure 4(C) and (D) varies significantly with the  $q_R$  value. The feasible workspace is bounded by the workspace singularities (red). For different values of redundant angle, one builds a stack of behavioural modes, each encoding the demonstrated motion pattern, allowing smooth interpolation or selective retrieval depending on task constraints. As a result of this modular arrangement, the safety and robustness properties of every individual behaviour mode guarantees a violation-free control policy for anthropomorphic redundant robots.

## Certified Execution

To model the robot control policy from joint trajectories, each trajectory is modelled a globally stable dynamical system (DS) in a one-shot manner (32). The proposed control policy, constructed as a composition of such stable dynamical systems, is resilient to both temporal and spatial perturbations. While small deviations are passively corrected, larger displacements, such as those moving the robot to a different joint space region, trigger a topology-aware feasibility analysis to assess whether the robot’s current aspect permits a constraint-compliant trajectory to the demonstrated goal.

Crucially, not all aspects of the joint space admit a valid inverse kinematic solution (IKS) to the demonstrated goal position. To address this, we develop certified factor-specific connectivity checks grounded in the analytical structure of  $\det \mathbf{J}(\mathbf{q})$ . There exist five factor types based on their branch types,  $(1, 0)[\geq 2, 0]$ ,  $(0, 1)[0, *]$ ,  $(1, 0)[2, > 0]$ ,  $(0, 0)[2, \geq 2]$ , and  $(1, 1)[0, *]$  (see Supplementary Methods section *Robot Categorisation*). Each irreducible factor of  $\det \mathbf{J}$  partitions the joint space into multiple regions, and feasibility is established only if the start and goal configurations belong to the same partition with respect to every factor. We leverage the global topological structure of the robot’s kinematic space into the control logic, a reasoning capability missing from standard learning-from-demonstration frameworks. Our method enables us to find connectivity in a deterministic and fast manner.

To leverage the certified kinematics properties while learning the demonstrated behaviour and formulating a control policy, the framework undertakes a multi-step process summarised in Figure 5, which depicts the implementation pipeline for executing a learned DS policy on a single robot. For each  $3R$  robot at a fixed value of  $q_R$ , Panel (A) groups the offline, one-time pre-processing steps to determine the robot category (details in the Supplementary Material, section “Robot Categorisation”). Panel (B) then corresponds to the planning stage, where we learn the aspect-specific nominal joint behaviour (see the Supplementary Material, section “Modelling the Robot Control Policy”). The resulting policy governs the reactive control loop shown in Panel (C), which combines connectivity checks with the attractor, constraint-violation monitoring, nominal DS policy execution, and track-cycle computations to ensure faithful replication of the demonstrated behaviour while respecting kinematic constraints (further details in “Modelling the Robot Control Policy”). The detailed pipeline, together with the wall-clock timings and time latency induced by

each step, is reported in Figure 5; these timings show that pre-processing is negligible compared to demonstration time and that both planning and reactive control incur only modest computational overhead, making the certified execution layer compatible with real-time control on the tested platforms.

Figure 6 illustrates certified execution and its aspect-based failure modes on a Category 1 arm whose workspace mixes regions with four and two IKS. From any chosen IKS, the controller executes the demonstration only if a feasible aspect connects start and goal; otherwise, it halts safely. For non-redundant 6R arms, failures occur when no admissible path exists or perturbations drive the state into joint-limit or singularity margins, triggering track cycles or a safe stop. The same logic extends to wrist-partitioned 7R arms (Figure 4), where tasks are declared infeasible if no redundant angle  $q_R$  yields a feasible aspect. A detailed discussion of the failure modes is presented in the Supplementary Materials, section ‘Robustness’.

## Experimental evaluation of “Demonstrate once, execute on many”

We experimentally validate the “demonstrate once, execute on many” principle on three industrial manipulators: a compact 6-DoF Duatic Dyna arm with tighter joint limits, a 7-DoF KUKA LWR iiwa7 with a wrist-partitioned architecture and moderate limits, and a 7-DoF Neura Robotics Maira M with longer links and more relaxed limits, yielding markedly different workspaces, proximity to limits, and feasible aspect layouts for the same end-effector task.

In Experiment 1, a human demonstrates a path resembling the letters of “SCIENCE” once per letter using a motion-capture glove and OptiTrack. The recorded trajectories are encoded as DS policies and replayed on the two 7-DoF arms, which reproduce the characters without joint-limit or singularity violations across multiple aspects and redundant angles (refer to Movies S3–S4).

In Experiment 2, we construct a mock multi-robot assembly line with three skills—pushing, pick-and-place, and throwing. Each skill is demonstrated once, encoded once as a DS in joint space, and executed on three arms without retraining. Pushing and throwing drive joints close to workspace boundaries and induce large excursions, while pick-and-place requires aspect queries

enforced by connectivity checks. The same policy is reused; only the kinematic embedding and certified execution layer change, keeping trajectories within feasible aspects and redirecting them along analytic track cycles near constraints. Figure 8 shows the three robots performing the three skills; full executions appear in Movie S6.

## DISCUSSION

Teaching robots through demonstration is becoming a natural and effective way for humans to communicate complex skills. As robotic platforms diversify, the ability to share learned behaviours across different systems, without reprogramming each one, becomes increasingly vital. This is the central goal of transfer learning from demonstration: to teach once and generalize broadly.

We propose that integrating behaviour learning with robot-specific constraint enforcement, by embedding analytic kinematic knowledge directly into the control policy, enables proactive avoidance of infeasible configurations without relying on post-hoc checks. Our framework advances Learning from Demonstration (LfD) by enabling generalizable and safe skill transfer across a wide class of robot architectures, through a principle we term kinematic intelligence. It separates user intent from robot morphology, making intent transferable across different robots.

Each demonstrated trajectory is modelled as a globally asymptotically stable dynamical system (DS), ensuring convergence to its target configuration. This stability at the most fundamental level yields robustness to temporal and spatial perturbations, without requiring reactive correction. As the control policy is incrementally built by adding more demonstrations, the stability of each DS ensures that the entire policy remains stable at every stage of learning.

Although behaviour is demonstrated in workspace, execution occurs in joint space. Therefore, reliable task replication requires embedding user intent in the joint space while accounting for robot-specific constraints. Our framework achieves this through: (1) a topological classification of singularities to categorise robot types, (2) connectivity analysis for assessing feasibility, and (3) constraint-aware policy synthesis. By encoding control policies within joint space aspects, regions bounded by singularity surfaces and joint limits, trajectories are guaranteed to remain within feasible boundaries, avoiding violations.

If a predicted trajectory nears a constraint, a near-boundary strategy redirects motion along a

safe path called the track cycle, rejoining the original trajectory once clear. These strategies are category-specific and embedded into the control logic, making the behaviour certifiable, guaranteed to be safe through bounded-complexity feasibility checks. All connectivity and constraint handling methods are derived analytically from the robot’s structure, ensuring that motion is grounded in the robot’s true capabilities.

The framework supports generalization through a transfer mechanism: joint configurations are sampled and mapped to the workspace, where the workspace model generates transfer trajectories. These are inverted into joint space, validated, and trimmed to retain only feasible segments. A novelty metric ranks these trajectories based on how much new information they contribute, those that expand the joint space coverage are incorporated into the control policy. This allows the system to generalize from limited examples without compromising safety.

Experiments validate the robustness of this approach on anthropomorphic robots with differing kinematics. A common demonstration is used to synthesize robot-specific joint-space controllers, enabling safe execution without retraining. As shown in Figure 4, the framework adapts to redundancy: it executes successfully for all redundant angles in feasible aspects, and correctly halts when a configuration becomes infeasible. This transparency in decision-making enhances user trust by making robot behaviour predictable and legible (33, 34).

While recent collaborative and industrial robots are cuspidal (e.g., ABB YuMi/GoFa, Fanuc CRX, Kinova Link 6) or incorporate non-spherical wrists (e.g., UR5), the conventional wrist-partitioned architecture is still widely adopted (e.g., Kuka LWR, Neura Maira, Dyna Arm) (25). Hence, studying wrist-partitioned non-cuspidal cases provides the essential foundation for developing certified, singularity-aware control policies, as it guarantees a clear separation of feasible regions and enables rigorous analytical treatment. In contrast, cuspidal manipulators permit continuous transitions between distinct IKS regions without crossing a singularity, which complicates connectivity analysis and can lead to nonsingular solution transitions (22, 35). Our focus was therefore on non-cuspidal architectures to establish provable safety and predictability before extending these principles to the more intricate cuspidal cases.

Extending the framework to cuspidal architectures is an important next step. Recent results show that all 3R manipulators admit *reduced aspects* (36), which provide an intrinsic subdivision of joint

space even in the presence of multiple IKS in an aspect. We envision leveraging these reduced aspects, together with the different types of admissible paths in cuspidal robots (21), to generalise our connectivity checks and track-cycle strategies to such architectures. This is particularly relevant for newer cuspidal designs such as ABB GoFa (37) and Fanuc CRX, where kinematic awareness is crucial for safe skill transfer. Explicit planning over redundant dimensions and smoother blending between nominal and corrective velocities could further improve path feasibility, collision avoidance, and motion fluidity near constraints. In our current implementation, the violation-check step monitors proximity to joint limits, singularities, and voids in workspace and triggers the corresponding track-cycle behaviour; in principle, self- and scene-collisions could be handled in exactly the same way by treating collision manifolds as additional joint-space boundaries and updating the track-cycle logic accordingly. Developing and validating this collision-aware, cuspidal-ready extension remains an important avenue for future work.

In summary, our framework provides a constraint-aware foundation for transferable robot learning. By embedding kinematic intelligence into the learning pipeline, it supports scalable, safe deployment across diverse platforms, bringing us closer to seamless integration of robotics in everyday life.

## MATERIALS AND METHODS

### Overview

This research aimed at developing a framework to transfer the demonstrated behaviour across a class of  $3R$  robots and redundant  $4R$  robots. The framework first models the user behaviour as is, agnostic to the robot and autonomously embeds the behaviour model with robot-specific kinematic intelligence, resulting in a control policy overseeing the safe and violation-free execution on a non-redundant robot. A modular arrangement of such non-redundant robot control policies, when composed together, enables behaviour execution on redundant robots.

## Behaviour modelling from demonstration

We begin by modelling the user-demonstrated behaviour in both the workspace  $\mathcal{X}$  and joint space  $\mathcal{Q}$  of the robot (see Figure 7(A1), (B), and (C)). Each user demonstration (trajectories in  $\mathcal{X}$ ) and its inverse-mapped versions in  $\mathcal{Q}$ , are individually embedded as globally asymptotically stable dynamical systems (DSs):  $\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x})$  for workspace trajectory, and  $\dot{\mathbf{q}} = \mathbf{g}(\mathbf{q})$  for joint trajectory. The DS formulation ensures that a trajectory from any starting point in the workspace or joint space converges to the demonstrated goal over time. Each DS is constructed in a *latent space*  $\mathcal{U}$ , a transformed coordinate system obtained via a bijective mapping  $\psi : \mathcal{Z} \rightarrow \mathcal{U}$ , where  $\mathcal{Z}$  is either  $\mathcal{X}$  or  $\mathcal{Q}$ . In latent space, the dynamics become (quasi)linear, simplifying stability guarantees. The learned dynamics in  $\mathcal{U}$  are then pulled back to the original space via inverse mapping,  $\psi^{-1}$ , producing non-linear but stable DSs in both spaces -  $\mathcal{X}$  and  $\mathcal{Q}$ .

To generalize across multiple demonstrations, all joint space trajectories that terminate in the same *aspect*  $\mathcal{A}_i$  are grouped together (see Figure 7(C)). Each resulting DS embedding  $\mathbf{g}_j^{(Q)}$  within this aspect is associated with a localized *region of influence* in aspect  $\mathcal{A}_i$  (see Figure 7(C)), defined as the set of joint configurations where  $\mathbf{g}_j^{(Q)}$  governs the motion. That is, for any  $\mathbf{q}$  in this region, the robot's behaviour is given by  $\dot{\mathbf{q}} = \mathbf{g}_j^{(Q)}(\mathbf{q})$ . An analogous partitioning is performed in the workspace when multiple demonstrations are available. The workspace partitioning spans the entire domain, in contrast to the joint space, where the partitioning is aspect-specific.

These regions are defined via *cones of influence* (see Figure 7(D1), and (D2)) in the latent space:

$$\mathcal{L}_j = \left\{ \mathbf{u} \in \mathcal{U} \mid \angle(\mathbf{u}^* - \mathbf{u}, \mathbf{v}_j) \leq \lambda_j \right\}, \quad \mathcal{R}_j = \psi_j(\mathcal{L}_j)$$

Here,  $\mathbf{u}^* = \mathbf{1}_n^T$  is the latent embedding of the goal, and  $\mathbf{v}_j$  is the vector from the start to goal. The angle  $\lambda_j$  controls the size of the neighbourhood influenced by the DS.

At inference time, the control law is given by:

$$\dot{\mathbf{q}} = \mathbf{g}_j^{(Q)}(\mathbf{q}), \quad \text{where } j = \arg \max \{P(\mathcal{R}_j) \mid \mathbf{q} \in \mathcal{R}_j\}$$

Priority  $P(\mathcal{R}_j)$  ensures disambiguation when regions overlap.

To prevent high-priority demonstrations from overriding lower-priority behaviours, the opening angle  $\lambda_j$  is adjusted using projection checks:

$$\lambda_j = \min \left\{ \angle \left( \psi_j^{-1}(\mathbf{q}_l) - \mathbf{1}_n^T, \mathbf{v}_j \right) \right\}_{\mathbf{q}_l \in \mathcal{Q}_l}$$

By combining such aspect-specific behaviours, we construct the full joint space policy. The process of modelling the joint space policy from multiple workspace demonstrations for the case in Figure 7 is presented in Movie S5.

For full derivations and algorithmic procedures, see Supplementary Methods: *Modelling the Robot Control Policy*.

## Embedding kinematic intelligence

To generalise user-demonstrated behaviours and ensure safe execution, we embed kinematic intelligence into the robot’s control policy. So far, we have modelled the behaviour in both workspace and joint space. While either model can act as a control policy, directly embedding constraints in the workspace model is challenging due to complex singularities (e.g., nodes, cusps) and non-linear boundaries formed by joint limits. Moreover, the workspace lacks homogeneity in position and orientation, complicating violation-free control.

Conversely, joint space singularities form continuous, differentiable  $C^\infty$  functions, and joint limits define a linear subspace, making joint space more suitable for embedding constraints while preserving user behaviour. We thus adopt the joint space model as the primary control policy, enriched via: (1) singularity-based robot categorisation and (2) incremental generalisation that transfers behavioural variability from workspace to joint space.

We introduce a novel, exhaustive categorisation of generic and non-generic non-cuspidal 3R robots into six categories. This reveals how motion becomes constrained, enabling targeted strategies for stable, feasible behaviour around constraints. We analyse the topology of joint space singularities, classifying them by loop structure and turning points to design locally stable, constraint-aware strategies.

Categories differ by whether branches loop, intersect, or fold, dictating directional constraints. Further details are in Supplementary Methods: *Robot Categorisation*.

If the joint space trajectory risks violating limits or singularities, we activate a near-boundary control policy to keep motion within the same aspect, bounded by joint limits and singularities, tailored per robot based on its singularity layout.

We define safety margins near aspect boundaries and monitor motion continuously. If a violation

is imminent, we switch to the near-boundary policy, which temporarily deviates along constraint-safe paths, termed the track cycle, before resuming nominal trajectory. Near-constraint strategies for all six categories are in Supplementary Methods: *Behaviour near Kinematic Constraints*. The framework integrates these strategies with deterministic connectivity checks of fixed computational complexity, ensuring stable and safe motion near constraints. These checks verify whether the configuration lies within a feasible aspect and can reach the goal without violating it; otherwise, execution halts, preventing unsafe behaviour. Details for all singularities in non-cuspidal 3R robots are in Supplementary Methods: *Reliable Execution*.

Embedding kinematic intelligence keeps the framework robot-agnostic during learning, but robot-specific during execution, allowing robots to safely and faithfully replicate user intent, even near constraints.

## Workspace to jointspace transfer for enhanced generalisation

To improve generalization of the joint-space control policy, we introduce an incremental *workspace-to-joint space transfer* mechanism. This begins by uniformly sampling the joint space  $Q$  to generate a grid of candidate joint configurations  $\{\mathbf{Q}\}^{\text{sample}}$ , which are then mapped to corresponding workspace positions  $\{\mathbf{x}\}^{\text{sample}}$  using forward kinematics. For each  $\mathbf{x}_i \in \{\mathbf{x}\}^{\text{sample}}$ , a *transfer trajectory*  $\mathbf{X}_i^{\text{transfer}}$  is generated using the workspace DS model  $B^{(X)}$  such that:

$$\mathbf{X}_i^{\text{transfer}} = \left\{ \mathbf{x}_t \mid \mathbf{x}_{t=0} = \mathbf{x}_i, \lim_{t \rightarrow \infty} \mathbf{x}_t = \mathbf{x}^* \right\}$$

These transfer trajectories are then inverted back into joint space, yielding a set  $\{\mathbf{Q}\}^{\text{transfer}}$  of candidate joint-space trajectories. Each trajectory is validated and trimmed to remove unsafe or constraint-violating segments using certified constraint checks. To ensure efficient expansion of the joint space model  $B^{(Q)}$ , we introduce a *novelty metric* that prioritizes trajectories contributing the most new information. The novelty of a trajectory is evaluated by computing its angular deviation in the latent space from existing demonstrations. Let  $\psi_j : Q \rightarrow \mathcal{U}$  denote the embedding function of the  $j^{\text{th}}$  learned DS, and let  $\mathbf{q}_t$  be a point on a transfer trajectory  $\mathbf{Q}^{\text{transfer}}$ . The angular deviation is given by:

$$\Theta_j(\mathbf{Q}^{\text{transfer}}) = \left\{ \angle \left( \psi_j^{-1}(\mathbf{q}_t) - \mathbf{1}^\top, \mathbf{1}^\top - \mathbf{0}^\top \right) \right\}_{t=1}^{|\mathbf{Q}^{\text{transfer}}|}$$

For an aspect  $\mathcal{A}_i$ , let  $B^{(\mathcal{A}_i)}$  be the corresponding aspect-specific behaviour. The novelty score of a trajectory is defined as:

$$\text{novelty}(\mathbf{Q}^{\text{transfer}}) = \min_j \left\{ \Theta_j(\mathbf{Q}^{\text{transfer}}) \mid \psi_j \in B^{(\mathcal{A}_i)} \right\}$$

The most novel trajectory is selected as:

$$\mathbf{Q}_{\text{novel}}^{\text{transfer}} = \arg \max_{\mathbf{Q}_i^{\text{transfer}} \in \{\mathbf{Q}\}^{\text{transfer}}} \text{novelty}(\mathbf{Q}_i^{\text{transfer}})$$

This most novel trajectory is then used to update the aspect-specific model  $B^{(\mathcal{A}_i)}$ , and by extension, the overall joint space behaviour  $B^{(Q)}$ . This ensures that the control policy becomes richer over time without compromising constraint adherence, stability, or model compactness. Further technical details, including latent space construction and trajectory validation steps, are provided in the Supplementary Methods section titled *Incremental Update by Workspace-to-Joint Space Transfer*.

## Controlling redundant robots

In robots with redundant degrees of freedom, those exceeding the minimal requirement to define end-effector poses, internal configurations can vary without affecting the external task. This inherent redundancy, while advantageous for optimising secondary criteria (e.g., obstacle avoidance, joint limits, or ergonomics), poses significant challenges for characterising and generalising robot behaviour. By leveraging Kinematic Intelligence, we address this by parameterising the redundant joint angle, denoted  $q_R$ , and reducing the complete 7R manipulator to a continuum of simpler, equivalent 3R chains, similar to one reported in (25). Each value of  $q_R$  yields a distinct robot architecture with well-defined kinematic properties, particularly concerning singularities and solution multiplicity.

This dimensionality reduction enables a structured encoding of robot behaviour as the redundant angle serves as an intrinsic coordinate along which motion solutions can be organised. For instance,

motion trajectories executed at different  $q_R$  values correspond to distinct internal configurations achieving the same external task. These configurations can be interpreted as behavioural modes, with  $q_R$  acting as a modulating variable. In non-cuspidal robots, where inverse kinematic solutions remain separated by singularities for each  $q_R$ , this parameterisation ensures that the operation mode is consistent and predictable.

This modular arrangement of behaviours allows redundancy resolution and behaviour generalisation across platforms. Crucially, it offers a modular, explainable framework for managing the solution null space of redundant manipulators, integrating geometric insight with practical applicability in control and learning. For applications demanding safety, repeatability, or human interaction, such an interpretable structure is essential for ensuring trustworthy robot behaviour. A detailed discussion of why classical numerical null-space strategies are insufficient in this context is provided in the Supplementary Materials, section *Issues with numerical path-planning strategies*.

## Supplementary materials

Materials and Methods

Figs. S1 to S24

Tables S1 to S3

Movies S1 to S6

## References and Notes

1. H. Ravichandar, A. S. Polydoros, S. Chernova, A. Billard, Recent Advances in Robot Learning from Demonstration. *Annual Review of Control, Robotics, and Autonomous Systems* **3** (1), 297–330 (2020), number: 1, doi:10.1146/annurev-control-100819-063206, <https://www.annualreviews.org/doi/10.1146/annurev-control-100819-063206>.
2. E. Gribovskaya, A. Billard, Combining Dynamical Systems control and programming by demonstration for teaching discrete bimanual coordination tasks to a humanoid robot, in *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (2008), pp. 33–40.
3. K. Kronander, A. Billard, Learning Compliant Manipulation through Kinesthetic and Tactile Human-Robot Interaction. *IEEE Transactions on Haptics* **7** (3), 367–380 (2014), doi:10.1109/TOH.2013.54.
4. A. L. P. Ureche, K. Umezawa, Y. Nakamura, A. Billard, Task Parameterization Using Continuous Constraints Extracted From Human Demonstrations. *IEEE Transactions on Robotics* **31** (6), 1458–1471 (2015), doi:10.1109/TRO.2015.2495003, <http://ieeexplore.ieee.org/document/7339616/>.
5. A. Pervez, A. Ali, J.-H. Ryu, D. Lee, Novel learning from demonstration approach for repetitive teleoperation tasks, in *2017 IEEE World Haptics Conference (WHC)* (2017), pp. 60–65, doi: 10.1109/WHC.2017.7989877.
6. N. Jaquier, *et al.*, Transfer learning in robotics: An upcoming breakthrough?: A review of promises and challenges. *The International Journal of Robotics Research* p. 02783649241273565 (2024), doi:10.1177/02783649241273565, <https://doi.org/10.1177/02783649241273565>.
7. B. D. Argall, S. Chernova, M. Veloso, B. Browning, A survey of robot learning from demonstration. *Robotics and Autonomous Systems* **57** (5), 469–483 (2009), number: 5, doi:10.1016/j.robot.2008.10.024, <https://linkinghub.elsevier.com/retrieve/pii/S0921889008001772>.

8. S. Calinon, F. Guenter, A. Billard, On Learning, Representing, and Generalizing a Task in a Humanoid Robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **37** (2), 286–298 (2007), doi:10.1109/TSMCB.2006.886952.
9. A. Shaw, J. Lee, J. Park, Constrained dynamic movement primitives for collision avoidance in novel environments. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2023).
10. L. Yang, M. Fischer, D. Kragic, Enhancing Learning from Demonstration with DLS-IK and ProMPs, in *International Conference on Automation and Computing (ICAC)* (2024).
11. E. Pignat, S. Calinon, Learning from demonstration using products of experts: Applications to manipulation and task prioritization. *The International Journal of Robotics Research* **41** (4), 373–394 (2022).
12. L. Bakker, V. Koltun, M. Toussaint, TamedPUMA: Safe and Stable Imitation Learning with Geometric Fabrics, in *Learning for Dynamics and Control (L4DC)* (2025).
13. B. Trabucco, M. Phielipp, G. Berseth, AnyMorph: Learning Transferable Policies By Inferring Agent Morphology, in *Proceedings of the 39th International Conference on Machine Learning*, K. Chaudhuri, *et al.*, Eds. (PMLR), vol. 162 of *Proceedings of Machine Learning Research* (2022), pp. 21677–21691.
14. C. Sferrazza, D.-M. Huang, F. Liu, J. Lee, P. Abbeel, Body Transformer: Leveraging Robot Embodiment for Policy Learning, in *8th Annual Conference on Robot Learning* (2024), <https://openreview.net/forum?id=0ce2215aJE>.
15. H. Klein, N. Jaquier, A. Meixner, T. Asfour, A Riemannian Take on Human Motion Analysis and Retargeting, in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2022), pp. 5210–5217, doi:10.1109/IROS47612.2022.9982127.
16. N. Makondo, B. Rosman, O. Hasegawa, Knowledge transfer for learning robot models via Local Procrustes Analysis, in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)* (2015), pp. 1075–1082, doi:10.1109/HUMANOIDS.2015.7363502.

17. P. Pastor, H. Hoffmann, T. Asfour, S. Schaal, Learning and generalization of motor skills by learning from demonstration, in *2009 IEEE International Conference on Robotics and Automation* (IEEE, Kobe) (2009), pp. 763–768, doi:10.1109/ROBOT.2009.5152385, <http://ieeexplore.ieee.org/document/5152385/>.
18. S. Schaal, J. Peters, J. Nakanishi, A. Ijspeert, Learning Movement Primitives, in *Robotics Research. The Eleventh International Symposium*, P. Dario, R. Chatila, Eds. (Springer Berlin Heidelberg, Berlin, Heidelberg) (2005), pp. 561–572.
19. M. Ewerton, G. Maeda, G. Kollegger, J. Wiemeyer, J. Peters, Incremental imitation learning of context-dependent motor skills, in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)* (2016), pp. 351–358, doi:10.1109/HUMANOIDS.2016.7803300.
20. D. H. Salunkhe, D. Chablat, P. Wenger, Trajectory planning issues in cuspidal commercial robots, in *2023 IEEE International Conference on Robotics and Automation (ICRA)* (2023), pp. 7426–7432, doi:10.1109/ICRA48891.2023.10161444.
21. D. H. Salunkhe, T. Marauli, A. Müller, D. Chablat, P. Wenger, Kinematic issues in 6R cuspidal robots, guidelines for path planning and deciding cuspidality. *The International Journal of Robotics Research* p. 02783649241293481 (2024), doi:10.1177/02783649241293481.
22. P. Wenger, J. El Omri, Changing posture for cuspidal robot manipulators, in *Proceedings of IEEE International Conference on Robotics and Automation* (IEEE, Minneapolis, MN, USA), vol. 4 (1996), pp. 3173–3178.
23. P. Wenger, Design of cuspidal and non-cuspidal robot manipulators, in *Proceedings of International Conference on Robotics and Automation*, vol. 3 (1997), doi:10.1109/ROBOT.1997.619284.
24. P. Wenger, Uniqueness domains and regions of feasible paths for cuspidal manipulators. *IEEE Transactions on Robotics* **20** (4), 745–750 (2004), doi:10.1109/TRO.2004.829467.
25. D. H. Salunkhe, S. Gupta, A. Billard, Cuspidal Redundant Robots: Classification of Infinitely Many IKS of Special Classes of 7R Robots. *IEEE Robotics and Automation Letters* **10** (12), 12509–12516 (2025), doi:10.1109/LRA.2025.3623011.

26. J. W. Burdick, A classification of 3R regional manipulator singularities and geometries. *Mechanism and Machine Theory* **30** (1), 71–89 (1995), doi:[https://doi.org/10.1016/0094-114X\(94\)00043-K](https://doi.org/10.1016/0094-114X(94)00043-K), <https://www.sciencedirect.com/science/article/pii/0094114X9400043K>.
27. P. Borrel, A. Liegeois, A study of multiple manipulator inverse kinematic solutions with applications to trajectory planning and workspace determination, in *Proceedings. 1986 IEEE International Conference on Robotics and Automation* (Institute of Electrical and Electronics Engineers, San Francisco, CA, USA), vol. 3 (1986).
28. P. Wenger, Classification of 3R Positioning Manipulators. *Journal of Mechanical Design* **120** (2), 327–332 (1998), doi:10.1115/1.2826976.
29. D. Paganelli, *Topological Analysis of Singularity Loci for Serial and Parallel Manipulators*, Ph.D. thesis, Universita di Bologna, Bologna, Italy (2008).
30. M. Asgari, I. Bonev, C. Gosselin, Singularities of ABB's YuMi 7-DOF robot arm. *Mechanism and Machine Theory* **205** (2025), doi:<https://doi.org/10.1016/j.mechmachtheory.2024.105884>.
31. A. J. Elias, J. T. Wen, Redundancy parameterization and inverse kinematics of 7-DOF revolute manipulators. *Mechanism and Machine Theory* **204**, 105824 (2024), doi:<https://doi.org/10.1016/j.mechmachtheory.2024.105824>.
32. S. Gupta, A. Nayak, A. Billard, [Under Review]Compact Oneshot Modelling of High Dimensional Demonstrations using Laplacian Eigenmaps. *IEEE Transactions on Robotics* **40** (2024).
33. A. Dragan, K. Lee, S. Srinivasa, Legibility and predictability of robot motion, in *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction, HRI '13* (IEEE Press) (2013), p. 301–308, doi:10.1109/HRI.2013.6483603.
34. J. Geldenbott, K. Leung, Legible and Proactive Robot Planning for Prosocial Human-Robot Interactions, in *2024 IEEE International Conference on Robotics and Automation (ICRA)* (2024), pp. 13397–13403, doi:10.1109/ICRA57147.2024.10611294.

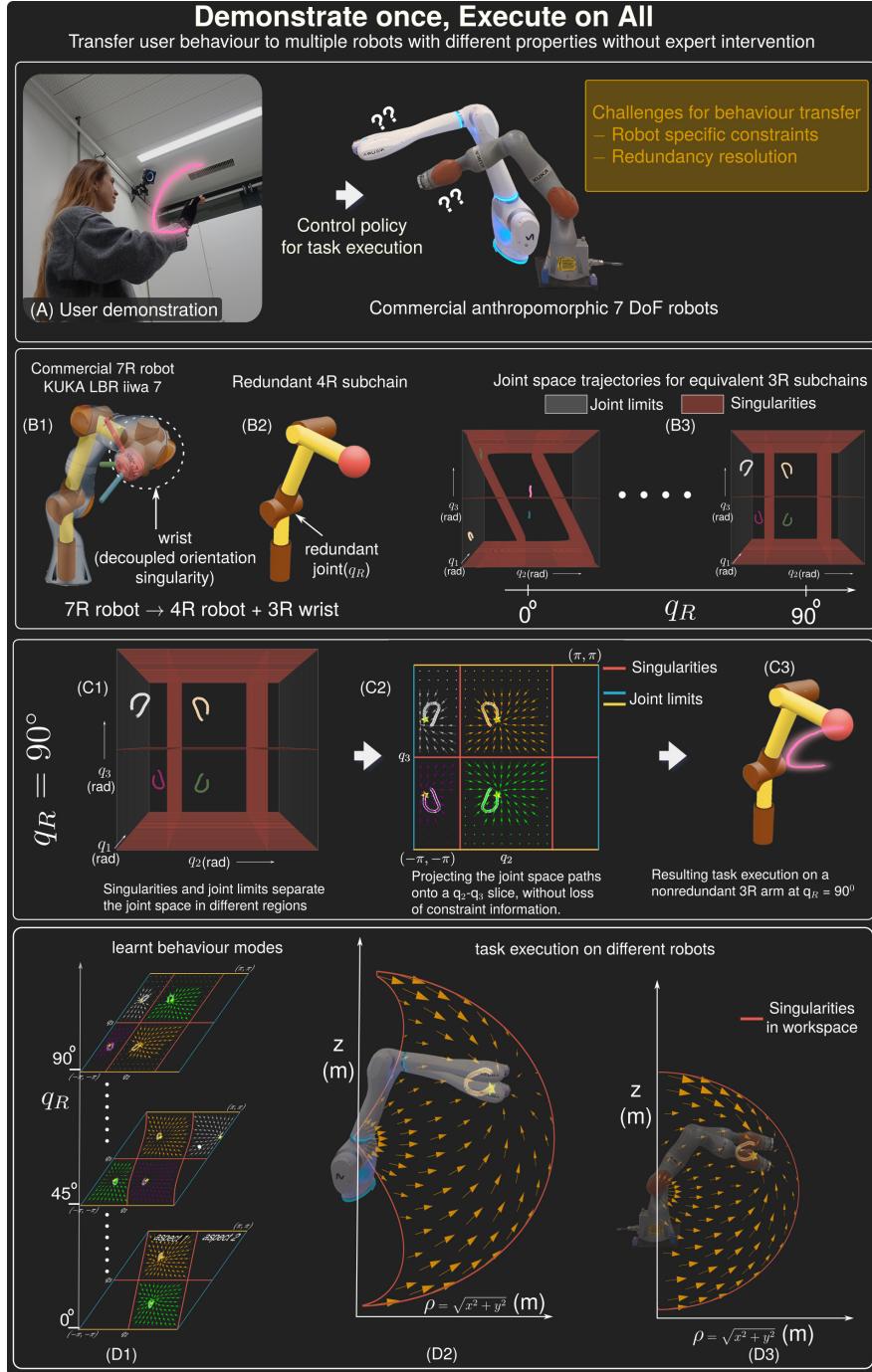
35. P. Wenger, D. Chablat, A Review of Cuspidal Serial and Parallel Manipulators. *Journal of Mechanisms and Robotics* **15** (4), 040801 (2022), doi:10.1115/1.4055677.
36. D. H. Salunkhe, C. Spartalis, J. Capco, D. Chablat, P. Wenger, Necessary and sufficient condition for a generic 3R serial manipulator to be cuspidal. *Mechanism and Machine Theory* **171**, 104729 (2022), doi:10.1016/j.mechmachtheory.2022.104729.
37. A. J. Elias, J. T. Wen, Path Planning and Optimization for Cuspidal 6R Manipulators. *arxiv* (2025), <https://arxiv.org/abs/2501.18505>.
38. J. Denavit, R. S. Hartenberg, A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. *Journal of Applied Mechanics* **22** (2), 215–221 (1955), doi:10.1115/1.4011045.
39. D. Kohli, J. Spanos, Workspace Analysis of Mechanical Manipulators Using Polynomial Discriminants. *Journal of Mechanisms, Transmissions, and Automation in Design* **107** (2), 209–215 (1985), doi:10.1115/1.3258710.
40. J. Burdick, A classification of 3R regional manipulator singularities and geometries, in *Proceedings. 1991 IEEE International Conference on Robotics and Automation* (1991), pp. 2670–2675 vol.3, doi:10.1109/ROBOT.1991.132033.
41. D. Pai, M. Leu, Genericity and singularities of robot manipulators. *IEEE Transactions on Robotics and Automation* **8** (5), 545–559 (1992).
42. J. W. Burdick, A classification of 3R regional manipulator singularities and geometries. *Mechanism and Machine Theory* **30** (1), 71–89 (1995), doi:[https://doi.org/10.1016/0094-114X\(94\)00043-K](https://doi.org/10.1016/0094-114X(94)00043-K), <https://www.sciencedirect.com/science/article/pii/0094114X9400043K>.
43. J. El Omri, P. Wenger, How to recognize simply a non-singular posture changing 3-DOF manipulator, in *Proc. 7th Int. Conf. on Advanced Robotics* (1995), pp. 215–222.
44. D. L. Pieper, *The Kinematics of Manipulators Under Computer Control*, Ph.D. thesis, Stanford University, USA (1968).

45. D. H. Salunkhe, *Robots cuspidaux : étude théorique, classification et applications aux robots commerciaux*, Ph.D. thesis, Ecole Centrale de Nantes, France (2023).
46. D. Chablat, R. Prébet, M. Safey El Din, D. H. Salunkhe, P. Wenger, Deciding Cuspidality of Manipulators through Computer Algebra and Algorithms in Real Algebraic Geometry, in *Proceedings of the 2022 International Symposium on Symbolic and Algebraic Computation*, ISSAC '22 (Association for Computing Machinery, Villeneuve-d'Ascq, France) (2022), p. 439–448, doi:10.1145/3476446.3535477.
47. K. Makarychev, Y. Makarychev, Certified Algorithms: Worst-Case Analysis and Beyond, in *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, T. Vidick, Ed. (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany), vol. 151 of *Leibniz International Proceedings in Informatics (LIPIcs)* (2020), pp. 49:1–49:14, doi:10.4230/LIPIcs. ITCS.2020.49.
48. S. M. Khansari-Zadeh, A. Billard, Learning Stable Nonlinear Dynamical Systems With Gaussian Mixture Models. *IEEE Transactions on Robotics* **27** (5), 943–957 (2011), number: 5, doi:10.1109/TRO.2011.2159412, <http://ieeexplore.ieee.org/document/5953529/>.
49. S. Salvador, P. Chan, FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space p. 11.

## Acknowledgments

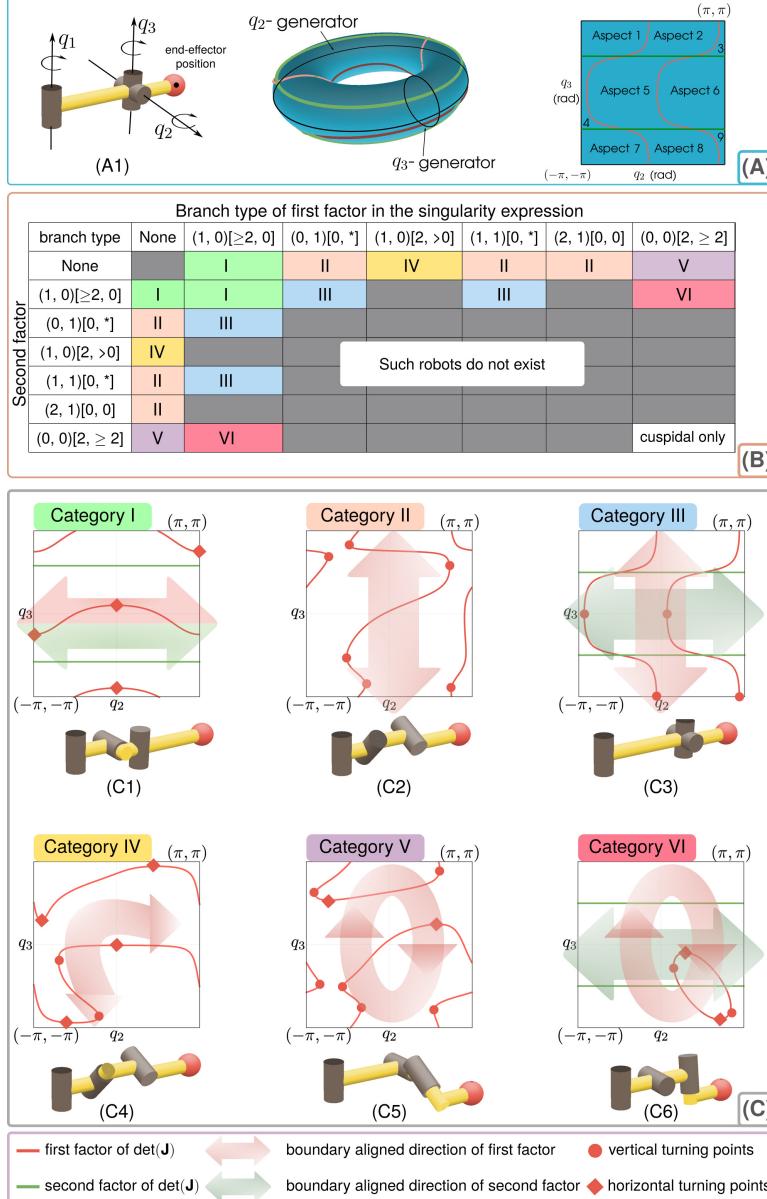
We thank *Dimitri Jacquemont* for their invaluable assistance in setting up and executing the robot experiments, and *Natacha Jeannot* for generously providing the demonstration data used in this work. We are also grateful to *Isabelle Derivaz-Rabii* for the essential administrative support during the project. We also used ChatGPT to assist with revising and refining the language in this paper.

**Funding:** This work was supported by EU project DARKO under Grant H2020 ICT-46-2020, and Horizon Europe under Grant 101070596, euROBIN.

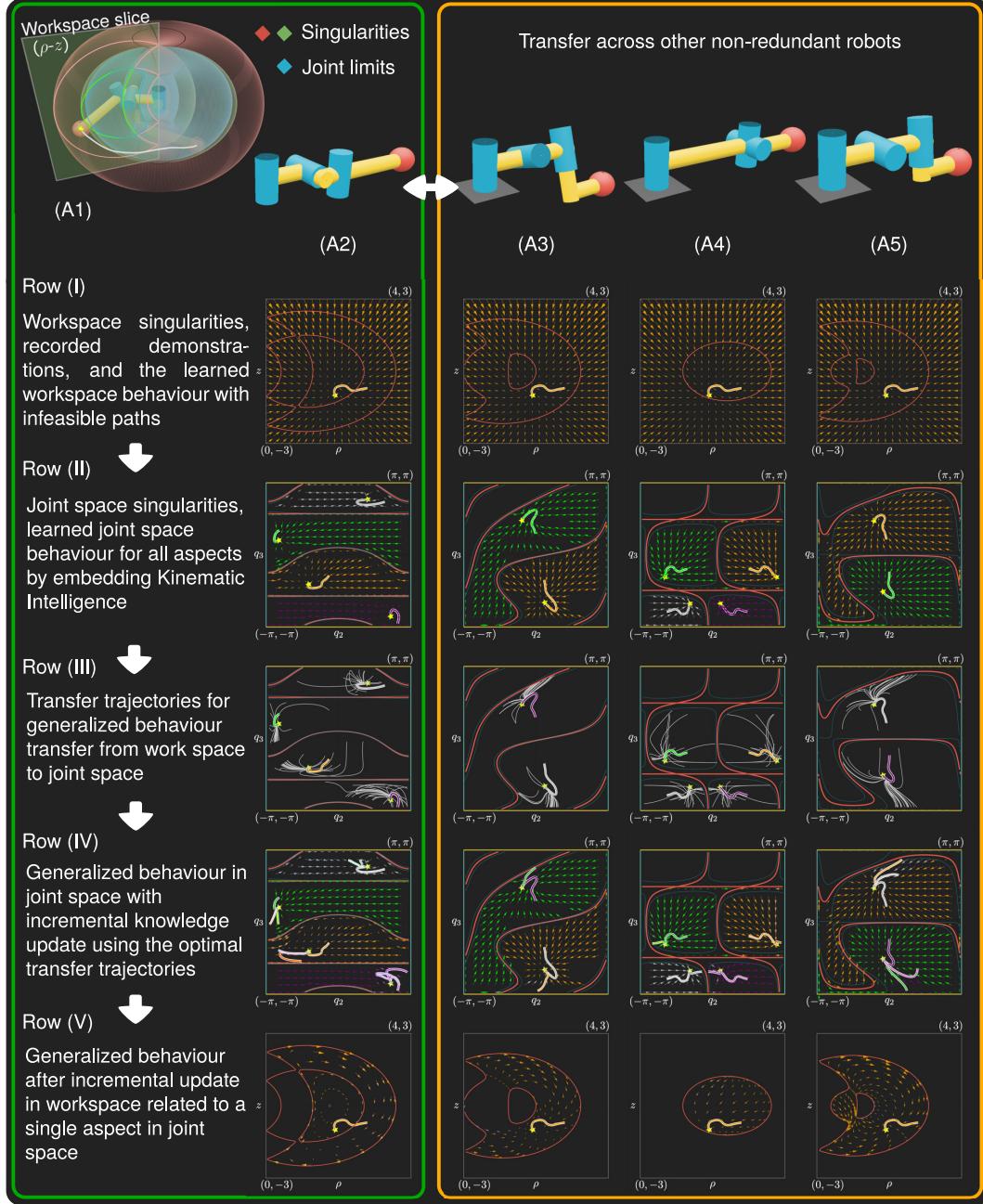


**Figure 1: Overview of kinematic-aware transfer of demonstrated behaviour to multiple robots**

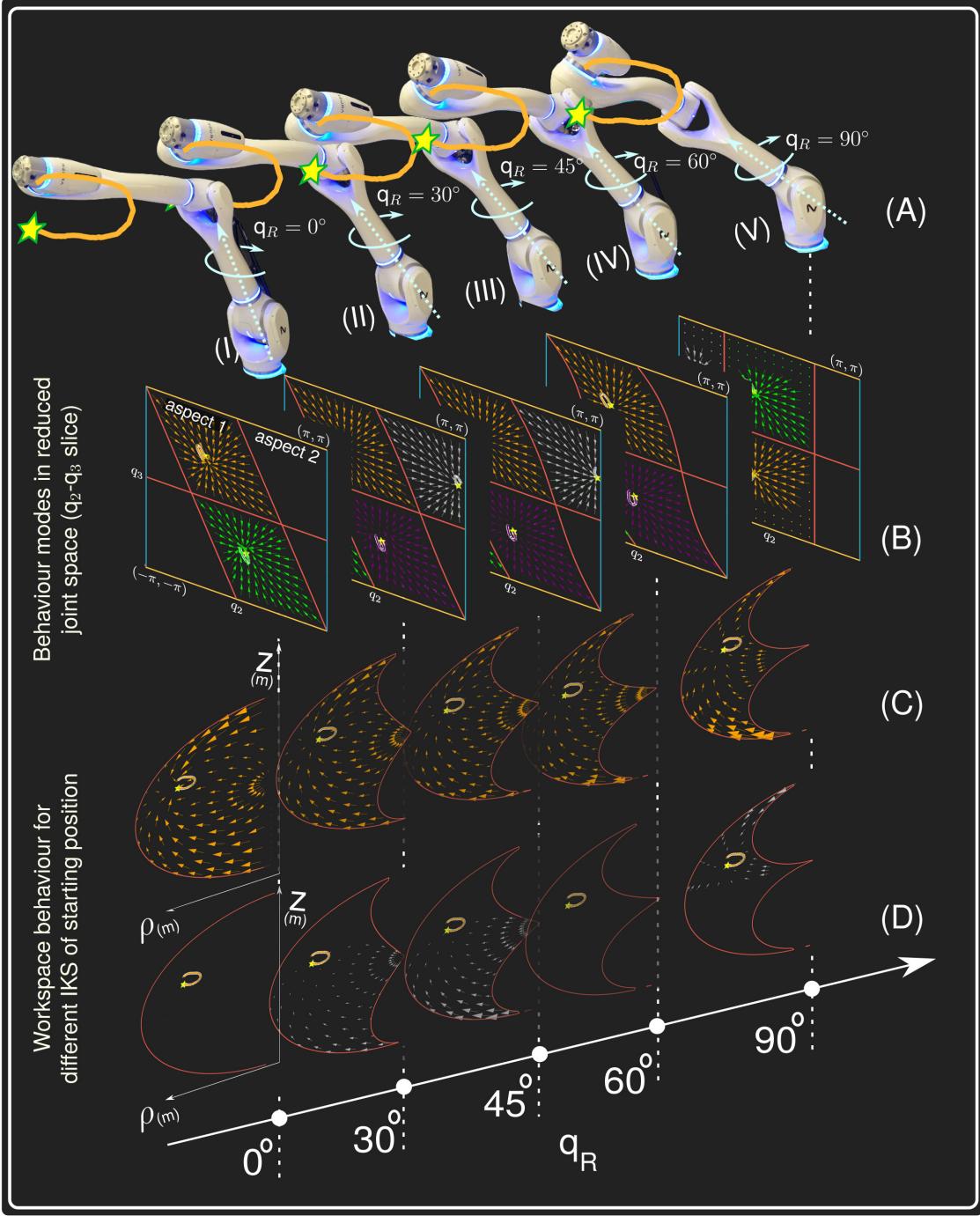
A user demonstrate a task (A) - challenge resides in embedding the control policy for easy transfer across multiple robots. We consider 7-DoF robots with [wrist-partitioned architecture](#) (B1), simplifying kinematics and reducing transfer to controlling the redundant 4-DoF subchain (B2). Redundancy, parameterized by the redundant joint angle  $q_R$ , identifies kinematic equivalence across 3-DoF non-redundant (3R) robots. (B3) Visualization of kinematic constraints (singularities and joint limits) for a specific kinematic, dividing joint space into four disjoint regions (C1). We learn a global control policy that generates stable subspace dynamics (C2), enabling violation-free task execution on a 3R robot (C3). Transfer across kinematics is achieved via policies parameterized by  $q_R$  (D1), with successful results on standard commercial robots (D2, D3).



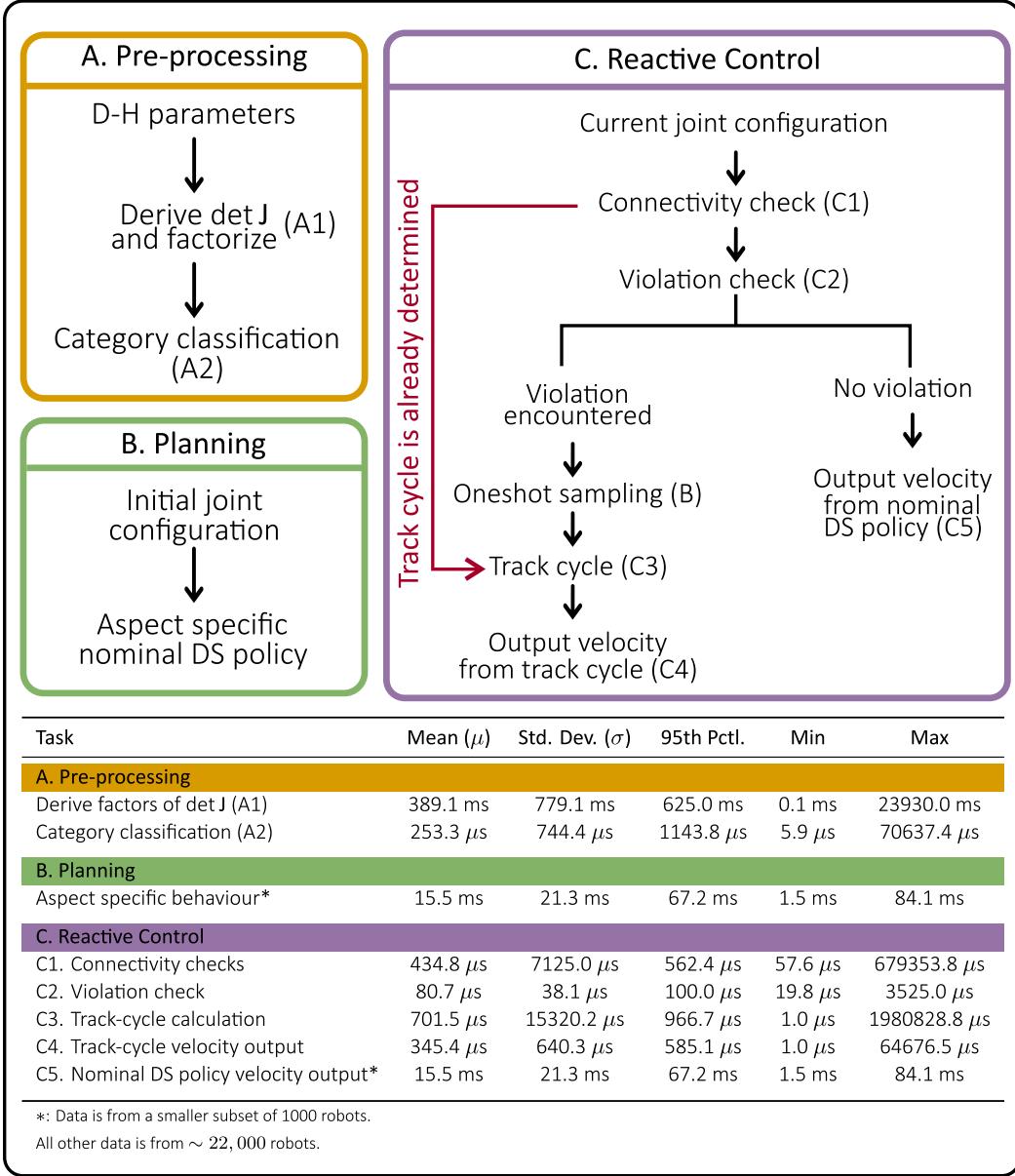
**Figure 2: Topological and differential properties of singularities as a basis for categorising 3R robots and defining boundary-aligned control actions.** To analyse the global properties of multiple robots, we study the different factors of the determinant of the Jacobian; their topological and differential properties give rise to six robot categories (B), and every non-cuspidal 3R robot falls into exactly one of these categories. Panels (C1)–(C6) show one representative robot per category, with its singularity curves and the associated *boundary-aligned directions* (arrows) drawn on the joint torus. Each set of arrows depicts the concrete control action used by the track-cycle controller when the state enters the safety band around that constraint: in Category I (C1), the track cycle traverses *horizontally* along the boundary; in Category II (C2), it traverses *vertically*; in Category III (C3), it switches between horizontal and vertical traversal depending on which singular branch is being violated; in Category IV (C4), it follows a primarily horizontal traversal but with additional turning-point considerations due to the more intricate branch geometry; in Category V (C5), the track cycle moves *clockwise* or *counterclockwise* around the torus along closed singular loops; and in Category VI (C6), it combines clockwise traversal around loops with horizontal drift along extended branches. In all cases, the nominal DS velocity is projected onto the depicted tangent direction so that motion remains within the same aspect and is steered along the boundary towards regions where re-entry into the interior is kinematically feasible.



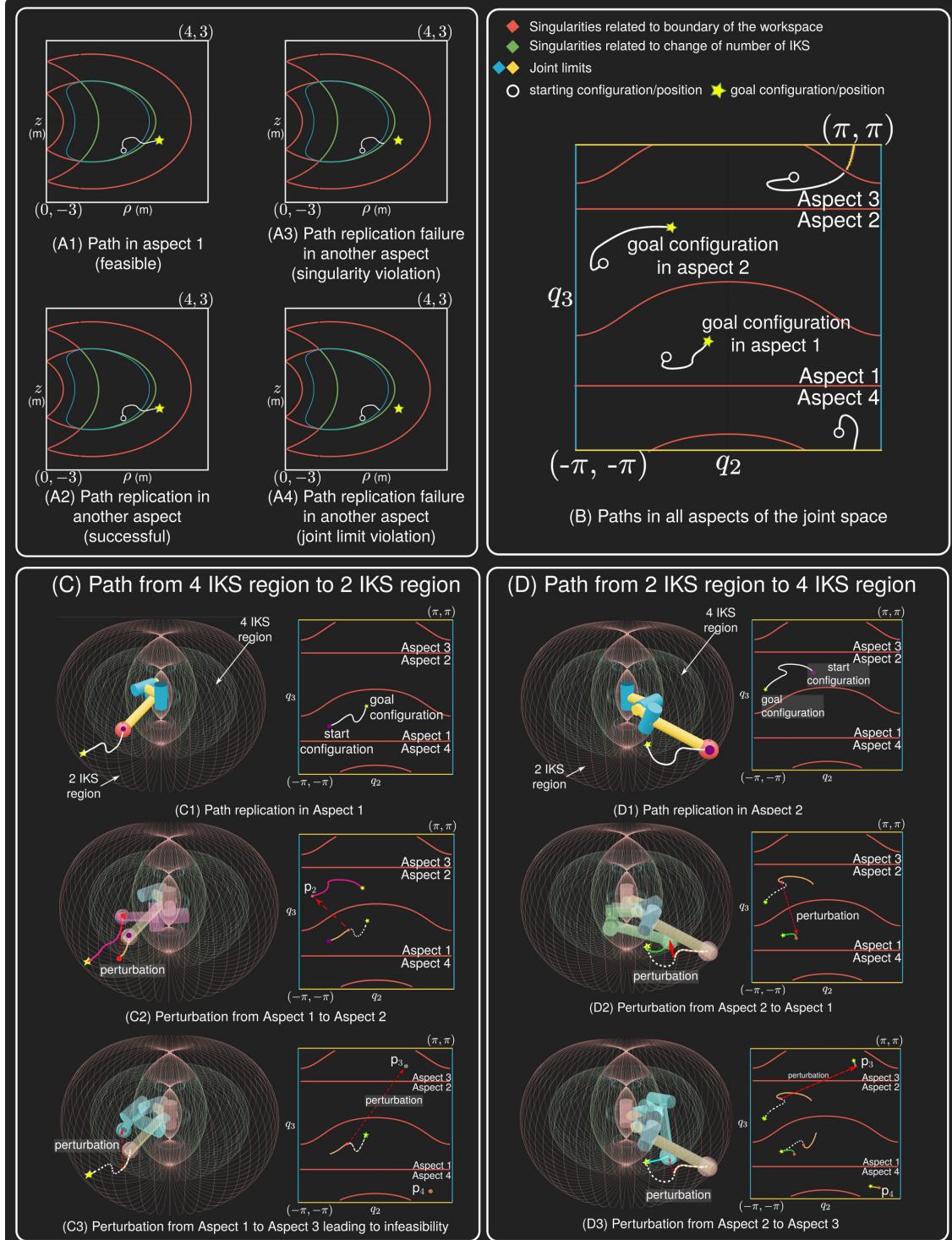
**Figure 3: Transfer of demonstrated behaviour to multiple non-redundant robots.** (A1) presents the Cartesian workspace of the 3R robot from (A2). The first column lists the steps from the learning framework, and each row reflects the status of these steps for four different robots - one each from Categories I, II, III, and IV - in columns (A2) through (A5), respectively. To visualise the user trajectory in the robot's workspace, we project it onto a  $\rho - z$  slice of the workspace (A1). Row (I) depicts the different workspace singularities in the  $\rho - z$  slice for the robots. Row (II) depicts the learnt control policy for every feasible aspect in the joint space of each robot. Row (III) depicts the set of transfer trajectories, used for incremental update of control policies. Row (IV) depicts the most novel transfer trajectory in each aspect and the updated behaviour. Row (V) depicts the resulting constraint-compliant workspace behaviour in a  $\rho - z$  slice for each robot.



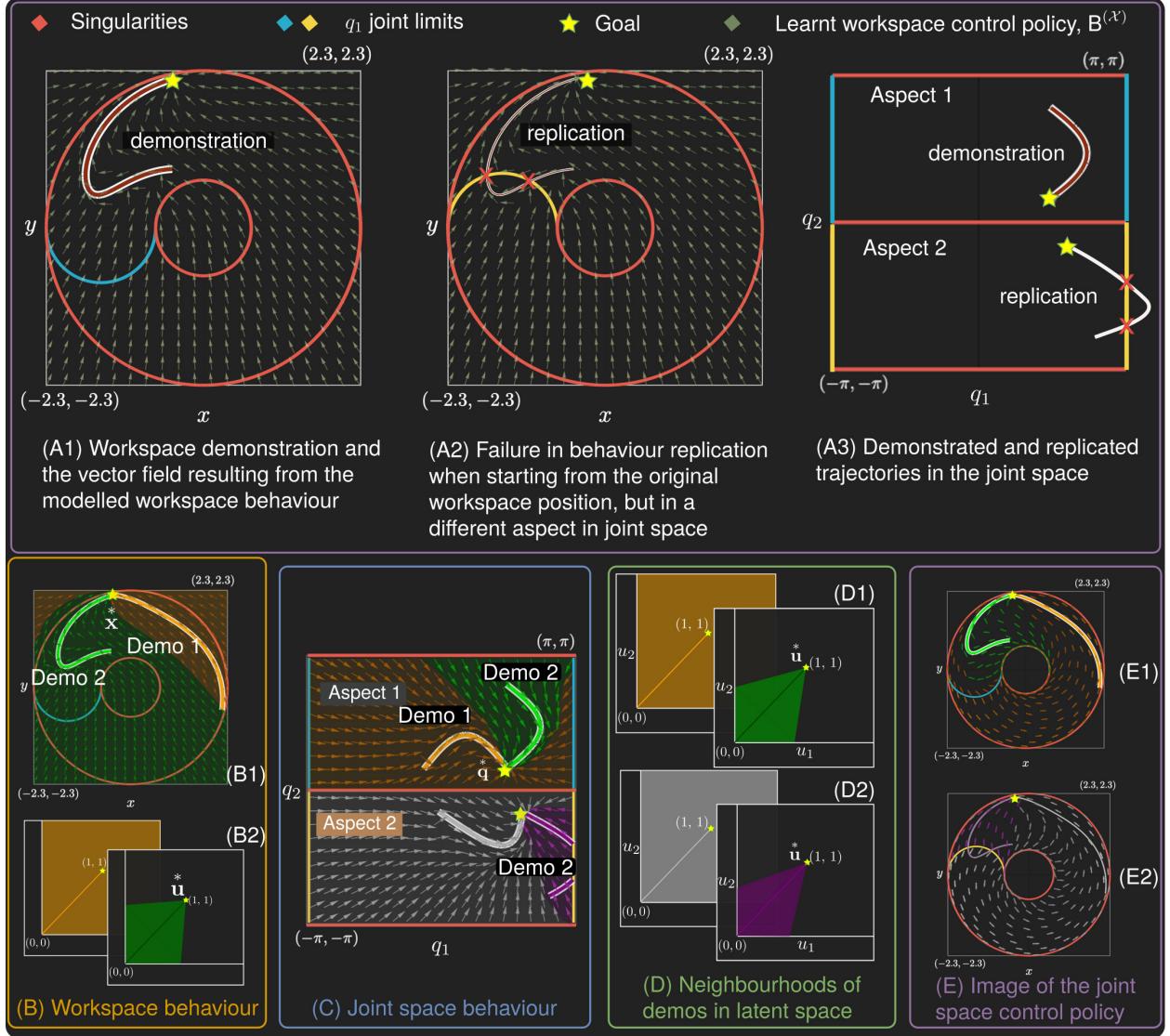
**Figure 4: Behaviour transfer to redundant robots through redundancy parametrisation** The behaviour of the redundant 4R subchain of a wrist-removed 7 DoF anthropomorphic robot can be characterised by discretising the values of the redundant angle  $q_R$ , which in this case is the third joint. Five values are picked for  $q_R - 0^\circ, 30^\circ, 45^\circ, 60^\circ$ , and  $90^\circ$  (A). Control policies for the equivalent non-redundant 3R subchains at these  $q_R$  values in the reduced  $q_2 - q_3$  slice of the joint space are presented in (B). (C) presents a  $\rho - z$  slice of the feasible workspace at each  $q_R$  value, where the robot (from (A)) is in a specific aspect of the joint space (orange). (D) presents the resulting workspace behaviour when the robot has a joint configuration that is a different solution of IK (silver).



**Figure 5: End-to-end DS-based policy execution on a single robot.** The figure illustrates the three main components of our framework for executing a learned dynamical system (DS) policy on a single robot and reports the wall-clock timings for each step of the process. (A) *Pre-processing*: an offline, one-time step that computes the kinematic structure, singularity layout, reduced aspects, and other quantities required for constraint-aware execution. (B) *Planning*: given an initial joint configuration, the framework selects the corresponding reduced aspect and synthesizes an aspect-specific reference behaviour compatible with the DS policy. (C) *Reactive control*: an online loop that combines connectivity checks, constraint-violation monitoring, nominal DS policy execution, and track-cycle computations to steer the robot while remaining within feasible regions safely. The accompanying wall-clock timings summarize the computational cost of each component, highlighting that the one-time offline pre-processing is very fast, and the online planning and reactive control steps are compatible with real-time execution.



**Figure 6: Certified execution for non-redundant robots** The figure analyses the workspace demonstration provided in Figure 3 (A1) on a 3R robot from Category I. The starting position has four IKS, whereas the goal position has two. (A1 - A4) visualise the workspace trajectory corresponding to each aspect, in a  $\rho - z$  slice of the workspace. Two of these trajectories - in Aspect 1 and Aspect 2 - reach the goal configuration without any violations (A1, A2). The trajectory in Aspect 3 encounters a singular configuration (A3), whereas that in Aspect 4 violates the third joint's limit (A4). (B) shows the joint space trajectories corresponding to the workspace demonstration, in each of the four aspects, and constraint violations encountered in Aspects 3 and 4. (C) presents the robot behaviour post learning the control policy and embedding kinematic intelligence. (C1) presents the replication when starting from Aspect 1. In (C2), the robot is suddenly perturbed to a point  $p_2$  in Aspect 2, yet the trajectory generated by the control policy converges at the goal configuration. (C3) shows a perturbation from Aspect 1 to Aspect 3, which leads to infeasibility. (D) presents the perturbation from Aspect 2 to Aspect 1 and Aspect 3, showing the robot's ability to handle such changes while maintaining its path towards the goal.



**Figure 7: An example of Learning and Replicating User-Intended Behaviour Across Workspace and Joint Space.** The top panel illustrates the modelling of a user-demonstrated trajectory in the workspace (A1) of a 2R robot, its corresponding joint space representation (A3), and the consequence of deploying the model from a different aspect, resulting in joint limit violation during replication (A2). The learned behaviour is encoded as a dynamical system (DS) in workspace and is aspect-agnostic unless explicitly constrained. The bottom panel demonstrates how multiple demonstrations are used to enrich the behaviour model. (B1) shows the DS embeddings and resulting workspace behaviour, while (B2) presents their latent space representations and cones of influence. (C) depicts the corresponding joint space trajectories, their trimming for constraint compliance, and the resulting aspect-specific joint space policies. (D1) and (D2) show the latent embeddings for joint space DSs in each aspect, with each DS's respective cone of influence. (E1) and (E2) display the resulting workspace behaviour when following the joint space control policy from initial configurations in Aspect 1 and Aspect 2, respectively.



**Figure 8: Experimental validation in a mock multi-robot automated assembly line: (A)**

Overview of the mock assembly-line setup, consisting of a conveyor belt (A1), workbench (A2), tabletop (A3), and basket (A4). (B) Human demonstration of the task: pushing the wooden block from the conveyor belt onto the workbench (B1); placing it on the tabletop (B2); lifting it into a throwing pose (B3); and throwing it into the basket (B4). (C) Three robots — Neura Maira M (C1), KUKA LBR iiwa 7 (C2), and Duatic DynaArm (C3) — are deployed to reproduce the demonstrated behaviour. Manipulation skills are learned and transferred across these robots, enabling consistent execution of each stage regardless of the robot arrangement.

**Author contributions:** Conceptualization, simulation, investigation, analysis, visualization, and writing: S.G, D.H.S. Resources, supervision, writing, and revision: A.B.

**Competing interests:** There are no competing interests to declare.