Screw-Wrench informed Impedance Variable Learning for Bimanual Manipulation of an Articulated Object

Kyungseo Park
Student ID : 2020-12295
Department of Mechanical Engineering
Seoul National University
erickun0125@snu.ac.kr

Abstract

Bimanual manipulation of articulated objects requires simultaneous satisfaction of kinematic constraints and regulation of inter-arm contact forces—capabilities that current imitation-based approaches that lack adaptive force modulation and rely on high-stiffness control. We introduce **SWIVL** (**S**crew-**W**rench informed **I**mpedance **V**ariable **L**earning), a hierarchical control framework bridging cognitive planning with physical execution. SWIVL contributes: (1) twist-driven impedance control via stable imitation vector fields that bypass nonlinear pose-error Jacobians; (2) screw axes based decompositions enabling independent compliance for bulk and internal motions; and (3) wrench-adaptive impedance variable learning via RL that suppresses excessive fighting forces. We evaluate SWIVL on BiarT, an SE(2) planar benchmark for bimanual articulated manipulation, demonstrating improved success rates and reduced internal forces compared to imitation learning baselines.

1 Introduction

Learning-based robotic manipulation has achieved strong performance in single-arm settings, particularly for structured pick-and-place tasks learned via demonstration. However, extending such approaches to **dual-arm manipulation of an articulated object** remains challenging. Coordinated bimanual interaction induces rich **inter-arm force coupling** and must satisfy complex **object-centric kinematic constraints**. Existing Learning-from-Demonstration (LfD) frameworks—typically grounded in high-stiffness position control with no explicit representation of object kinematics—often generate unstable motions and large internal forces when multiple arms physically interact with a shared object.

Cognitive vs. Physical Intelligence

To reason about this challenge, we distinguish between two complementary aspects of robot decision-making: **Cognitive Intelligence** and **Physical Intelligence**.

Cognitive Intelligence addresses high-level task understanding through semantic reasoning and planning. Modern VLA-based robot foundation models excel at interpreting goals and decomposing instructions via visual-language understanding based on VLM backbones. However, language operates as an abstracted symbolic representation that lacks the resolution needed for precise physical interaction—making it difficult to specify fine-grained motor commands for contact-rich manipulation such as force modulation or coordinated compliance control.

Physical Intelligence, in contrast, ensures safe and stable execution through explicit modeling of dynamics and kinematic principles. This includes regulating contact forces, satisfying geometric constraints, and generating dynamically consistent motions. This dimension remains underexplored in learning-based manipulation.

Most existing work advances Cognitive Intelligence through robot foundation models trained on cross-embodiment datasets. However, this approach introduces three limitations for contact-rich manipulation. First, cross-embodiment training cannot standardize low-level physical signals—wrench feedback, reference frames, and kinematic constraints vary across platforms. Second, imitation learning reproduces demonstrated trajectories without exploring contact dynamics, lacking the adaptive force modulation that reinforcement learning provides. Third, policies output poses executed via high-stiffness control, that can generate excessive forces and constraint violations. These limitations compound in bimanual articulated object manipulation, where force coupling and closed-chain constraints demand physical reasoning beyond current cognitive approaches. This mismatch motivates methods that **bridge high-level cognitive policies to low-level physically grounded control**.

Problem Focus and Scope

We focus on developing a low-level control stack for bimanual manipulation of an articulated object. Our formulation assumes access to (1) 6-axis wrench measurements at each end-effector via wrist-mounted force/torque sensors, and (2) the screw axes of an articulated object. While not all manipulation scenarios provide such information, these assumptions are satisfied in structured domains—including repetitive assembly, industrial workflows that represent high-value deployment scenarios for dual-arm systems. Recent advances in geometric perception modules—screw-axis estimation (e.g., Screw-Splatting) further expand the applicability of our framework.

In this setting, we use high-level planners (VLA, Imitation Learning Policies, Teleoperation Interfaces) provide motion intentions as desired end-effector poses, without explicitly reasoning about object's physical information or inter-arm force interactions. Our hierarchical control stack transforms these commands into physically feasible bimanual coordination that satisfies kinematic constraints, suppresses internal forces, and ensures compliant interaction—without requiring the high-level planner to reason about any of these physical details.

1.1 Our Approach : SWIVL

To address these problems, we introduce **SWIVL** (**S**crew-**W**rench informed **I**mpedance **V**ariable **L**earning). SWIVL enables safe bimanual manipulation with object-aware decomposed impedance control via screw axes and adaptive impedance variable modulation via wrench feedback. This work makes the following contributions:

1. **Twist-Driven SE(3) Impedance Control via Stable Imitation Vector Fields**: Bypass the nonlinear pose-error Jacobian inherent in SE(3) impedance control by incorporating pose errors directly into reference twists, enabling geometrically consistent compliance even under large trajectory deviations.

2. **Screw Axes-Decomposed Twist and Wrench Spaces**: Orthogonal projection operators structurally partition control into internal (joint articulation) and bulk (object transport) components, and dually decompose wrenches into internal forces and bulk forces, enabling independent compliant behavior for each object kinematics aware subspace.

3. **Wrench-adaptive Impedance Variable Learning**: A reinforcement learning policy modulates impedance parameters conditioned on object screw axes and real-time wrench feedback streamed from wrist-mounted force-torque sensors, learning to suppress harmful fighting forces while maintaining compliant trajectory tracking across diverse joint types.

We validate SWIVL on SE(2) benchamarks with articulated objects spanning two joint types: revolute and prismatic. Our results demonstrate that SWIVL achieves higher success rates and lower internal forces compared to imitation learning baselines, highlighting the importance of explicit physical modeling for bimanual manipulation of articulated objects.

2 Related Work

2.1 Learning Paradigms for Manipulation

Imitation Learning and Behavior Cloning. Learning from Demonstration (LfD) has become a dominant paradigm for acquiring manipulation skills. Behavior Cloning directly maps observations to actions through supervised learning, while recent methods employ generative models—Diffusion Policy Chi et al. [2025] for denoising-based action generation and ACT Zhao et al. [2023] for transformer-based action chunking—to improve temporal consistency. However, these approaches inherit two fundamental limitations: they reproduce demonstrated trajectories without exploring contact dynamics, and they execute learned policies via high-stiffness position control that cannot accommodate force interactions or kinematic constraints. These limitations become critical in bimanual manipulation where constraint satisfaction and force regulation are essential.

Vision-Language-Action Models. Cross-embodiment foundation models such as RT-1 Brohan et al. [2022], RT-2 Brohan et al. [2023], π_0 Black et al. [2024], and Octo Octo Model Team et al. [2024] leverage large-scale datasets like Open-X Embodiment O'Neill et al. [2024] for broad semantic understanding and impressive generalization to novel objects and instructions. These models advance Cognitive Intelligence—interpreting goals, decomposing tasks, and generating motion intentions. However, cross-embodiment training cannot standardize Physical Intelligence: wrench feedback, reference frame conventions, and kinematic constraints vary fundamentally across robot platforms and are absent from training data. This limitation motivates our approach of decoupling high-level cognitive planning from low-level physically grounded execution.

Reinforcement Learning for Contact-Rich Manipulation. Reinforcement learning enables acquisition of contact-rich skills through physical exploration—capabilities absent in imitation learning. Deep RL has achieved remarkable results in dexterous in-hand manipulation Andrychowicz et al. [2020] and tool use by optimizing physical objectives through environmental interaction. However, most RL formulations target single-task settings with dense, task-specific rewards. In contrast, SWIVL leverages RL to learn task-agnostic impedance modulation that generalizes across manipulation scenarios, using wrench feedback and object geometry as the primary learning signals.

2.2 Force-Compliant Control

Compliant Control and Force Regulation. Operational space control Khatib [1987] and impedance control Hogan [1985] provide foundations for compliant manipulation, regulating motion-force relationships through virtual mass-spring-damper dynamics. Hybrid force/position control Raibert and Craig [1981] extends this by decomposing task space into position-controlled and force-controlled subspaces. However, these classical frameworks face key limitations: mathematical formulations natural in Euclidean space require nonlinear Jacobians for SE(3) extension; fixed impedance parameters preclude adaptation to varying contact dynamics; and hybrid control requires a priori specification of control modes per axis. SWIVL addresses these through learned impedance modulation conditioned on wrench feedback, while our twist-driven formulation sidesteps SE(3) pose-error complexities.

Learning Force-Aware Manipulation. Recent work incorporates force sensing into learned policies through force-conditioned architectures and tactile-guided manipulation Lee et al. [2019], Suomalainen et al. [2022]. These approaches demonstrate improved performance on contact-rich tasks but typically learn task-specific force patterns that do not transfer across scenarios. SWIVL instead learns generalizable force-compliant behavior by decomposing wrenches into productive and non-productive components based on object kinematics, enabling task-agnostic force regulation.

2.3 Hierarchical Integration of Cognitive and Physical Control

Residual Reinforcement Learning. Residual RL bridges imitation and reinforcement learning by augmenting frozen behavior-cloned policies with learned corrective actions Johannink et al. [2019], Ankile et al. [2025]. This approach preserves demonstrated behavior while adapting to distribution shifts through online interaction. However, existing residual methods focus on single-arm manipulation without addressing the force coupling and constraint satisfaction required for bimanual coordination. SWIVL takes a different approach: rather than correcting a frozen policy, it learns a standalone low-level controller that operates beneath arbitrary high-level planners.

Whole-Body Control. Whole-body control provides a principled framework for translating high-level task commands into physically feasible motions on complex robotic systems. Recent work in humanoid control employs MPC and RL-trained low-level policies He et al. [2025, 2024] to handle the physical interactions arising from full-body dynamics—tracking commands from high-level planners such as VLAs and imitation learning policies while maintaining balance and stability. SWIVL applies this hierarchical philosophy to a different physical challenge: bimanual manipulation of articulated objects, where inter-arm force coupling and closed-chain kinematic constraints create complex physical interactions analogous to whole-body contact dynamics. Just as humanoid controllers learn to regulate ground reaction forces and joint coordination, our RL-trained impedance modulation policy learns to regulate internal forces and constraint satisfaction.

3 Method

We present **SWIVL** a hierarchical control framework that bridges high-level cognitive planning with physically grounded bimanual execution. SWIVL consists of three key components: (1) a Stable Imitation Vector Field based **Reference Twist Field Generator** that transforms discrete high-level waypoints into dense, continuous vector fields defined over the entire task space, (2) a Screw Axes Decomposition based **Twist-driven Impedance Controller** that enables practical impedance control with pose error and twist error, and (3) a Reinforcement Learning based **Wrench-feedback and Object-conditioned Impedance Variable Learning Policy** that modulates compliance in a physically feasible manner.

3.1 Problem Setup and Key Challenges

We address **bimanual manipulation of articulated objects** where two robot arms cooperatively manipulate a shared object with k internal degrees of freedom. Let $T_{sb_l}, T_{sb_r} \in \mathrm{SE}(3)$ denote the end-effector poses and ${}^s\mathcal{V}_{b_l}, {}^s\mathcal{V}_{b_r} \in \mathbb{R}^6 \cong \mathfrak{se}(3)$ denote the end-effector twists in the space frame. For articulated objects, $\mathbf{q}_{\text{obj}}, \dot{\mathbf{q}}_{\text{obj}} \in \mathbb{R}^k$ are the joint positions and joint velocities, and the kinematic structure is characterized by the spatial Jacobian $\mathbf{J}_s(\mathbf{q}_{\text{obj}}) \in \mathbb{R}^{6\times k}$, whose columns correspond to the *instantaneous* spatial screw axes determined by the current robot and joint configuration.

Holonomic Constraint. Since both end-effectors rigidly grasp the object, their relative motion is constrained to match the motion generated by the object's internal joints. This relationship is expressed in the spatial twist domain as:

$$ {}^s\mathcal{V}_{b_l} - {}^s\mathcal{V}_{b_r} = \mathbf{J}_s(\mathbf{q}_{\text{obj}})\dot{\mathbf{q}}_{\text{obj}}, \tag{1} $$

where

- ${}^s\mathcal{V}_{b_l}, {}^s\mathcal{V}_{b_r} \in \mathbb{R}^6$ denote the spatial twists of the left and right end-effectors, respectively.
- $\dot{\mathbf{q}}_{\text{obj}} \in \mathbb{R}^k$ represents the joint velocity vector.

Goal. Given high-level waypoints $\{T_{sd_i}[\tau]\}_{\tau=0}^{H}$ from cognitive planners (VLAs, behavior cloning policies, or teleoperation) that lack physical awareness, develop a low-level control stack that: (i) generates dense, stable reference motions, (ii) satisfies kinematic constraints compliantly, and (iii) minimizes fighting forces while tracking references.

Key Challenges. This problem presents four interrelated challenges that directly motivate SWIVL's architectural components:

C1. **From Discrete Poses to Dense Twist References.** Modern learning-based policies output **action chunks**—sequences of waypoints enabling temporal consistency and multi-step reasoning. However, these are inherently **sparse** (discrete snapshots vs. continuous control) and **open-loop** (no corrective feedback when deviations occur due to inter-arm forces). More fundamentally, there exists a **pose space vs. twist space mismatch**: kinematic constraints and impedance motions are linear in $\mathfrak{se}(3)$ but nonlinear in $\mathrm{SE}(3)$.

C2. **Compliant Constraint Satisfaction.** A natural approach is to structurally enforce Eq. (1) in the action space. However, this **hard constraint approach is brittle**: any small error in $\mathbf{J}_s$ due to perception noise, calibration errors, or object model mismatch directly translates to physically infeasible commands. When executed via high-stiffness control, these generate large fighting forces risking grasp slippage or hardware failure.

C3. **SE(3) Impedance Complexity.** Impedance control on SE(3) involves a nonlinear Jacobian $J_{\mathcal{E}}$ that couples rotation and translation in configuration-dependent ways (Appendix B). Commonly used approximations—small orientation errors and diagonal stiffness matrices—fail in our scenario due to large trajectory deviations from constraint violations and non-diagonal stiffness requirements for motion decomposition.

C4. **Bulk Motion Ambiguity.** Directly tracking raw end-effector references is ill-suited for object-aware manipulation. Instead, the control objective effectively requires decoupling into *internal motion* and *bulk motion* . However, a fundamental conflict arises when high-level planners generate kinematically inconsistent commands, the bulk motion (orthogonal part to internal joint motions) components of $\mathcal{V}_l^{\text{ref}}$ and $\mathcal{V}_r^{\text{ref}}$ do not match. This ambiguity manifests as **inter-arm force coupling**: wrench components orthogonal to screw axes directly reflect bulk motion disagreement.

Framework Scope. We develop the theoretical framework in SE(3) with multi-DoF articulated objects for generality, while experimental validation is conducted in SE(2) with 1-DoF objects to isolate core challenges.

Notation. We use $\{s\}$ for spatial frame, $\{b_i\}$ for body frame of end-effector $i \in \{l, r\}$, $\{d_i\}$ for desired frame, T_{ab} for transformation from $\{b\}$ to $\{a\}$, and ${}^a\mathcal{V}_b$ for twist of frame $\{b\}$ expressed in $\{a\}$.

3.2 Architecture Overview

SWIVL adopts a four-layer hierarchical architecture (Figure 1) that decouples high-level reasoning from low-level physical interaction:

- **Layer 1 (High-Level Policy)**: VLA, behavior cloning, or teleoperation providing sparse waypoints $\{T_{sd_i}[\tau]\}_{\tau=0}^{H}$ at low frequency ($\sim$10Hz).
- **Layer 2 (Reference Twist Field Generator)**: Transforms sparse waypoints into dense, closed-loop reference twists $\mathcal{V}_i^{\text{ref}}$ with stability guarantees (**C1**).
- **Layer 3 (Impedance Variable Modulation Policy)**: RL policy π_θ that modulates impedance variables based on object geometry and wrench feedback (**C3, C4**).
- **Layer 4 (Screw Axes-Decomposed Impedance Controller)**: Executes compliant control with independent regulation of bulk and internal motions (**C2, C3**).

The core innovation lies in the tight integration of Layers 2–4, which together enable physically grounded, force-compliant bimanual manipulation underneath arbitrary high-level planners.

3.3 Stable Imitation Vector Field (Layer 2)

This layer addresses **C1** by bridging the gap between discrete high-level waypoints and continuous low-level control. High-level policies provide sparse waypoints at low frequency ($\sim$10Hz), while the low-level controller requires smooth, dense reference trajectories at high frequency ($\sim$100Hz) with pose-error based corrective fields.

A key insight is the **pose space vs. twist space mismatch**: high-level planners naturally output poses $T \in \mathrm{SE}(3)$, but kinematic constraints and impedance control are fundamentally more tractable in the twist space $\mathfrak{se}(3)$. Although Equation (1) presents a linear relationship in the velocity domain, the fundamental geometric constraint resides in the configuration space $SE(3)$. For a general articulated object defined by a kinematic chain of k joints, the relative pose between the two end-effectors is governed by the Product of Exponentials (POE) formula:

$$T_{sb_l}^{-1} T_{sb_r} = \left(\prod_{i=1}^{k} e^{[\mathcal{S}_i] q_i} \right) M_{\text{obj}} \,, \tag{2}$$

where:

- $T_{sb_l}, T_{sb_r} \in SE(3)$ represent the current spatial poses of the left and right end-effectors.
- $\mathcal{S}_i \in \mathbb{R}^6$ denotes the spatial screw axis of the i-th joint at the zero configuration.

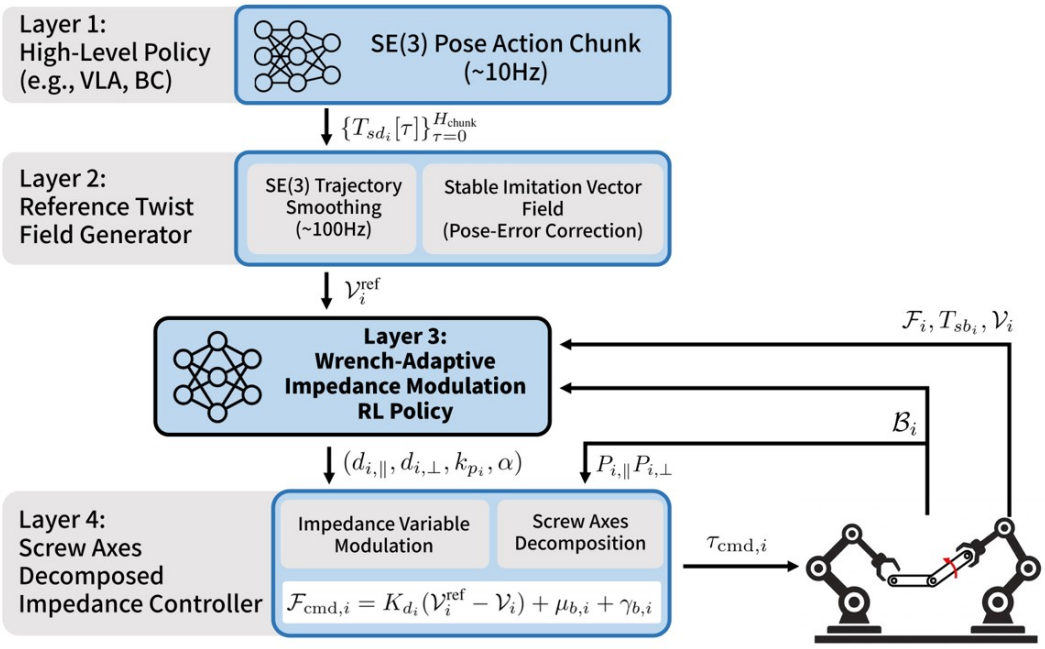


Figure 1: **SWIVL Architecture.** Four-layer hierarchy bridging cognitive planning with physical execution. High-level waypoints are transformed into stable reference twists (Layer 2), which are tracked by a screw-decomposed impedance controller (Layer 4) with learned compliance modulation (Layer 3). Wrench feedback enables adaptive force regulation.

- q_i is the position of the i-th joint.
- $M_{\text{obj}} \in SE(3)$ is the constant relative transformation between the end-effectors at the zero configuration ($\mathbf{q}_{\text{obj}} = \mathbf{0}$).

Nonlinear constraints in SE(3) manifold (Eq. (2)) motivates transforming pose commands into twist-space representations where constraints are linear in vector space.

The Reference Twist Field Generator performs three steps:

Step 1: SE(3) Trajectory Smoothing. To address the sparsity gap, we perform smooth interpolation in SE(3) to obtain dense desired trajectories at the control frequency $\Delta t_{LL} \ll \Delta t_{HL}$:

$$\{T_{sd_l}(t), T_{sd_r}(t)\}_{t=0}^{H_{LL}}, \tag{3}$$

where H_{LL} is the smoothed trajectory horizon. We use SLERP for rotations and cubic splines for translations. Both schemes are differentiable in time, inducing continuous position and rotation trajectories $p_i^{\text{des}}(t)$ and $R_i^{\text{des}}(t)$.

Step 2: Body Twist Computation. For an interpolated trajectory $T_{sd_i}(t) = \begin{bmatrix} R_{sd_i}(t) & p_{sd_i}(t) \\ 0 & 1 \end{bmatrix}$, we compute the desired body twist by differentiation:

$$\mathcal{V}_i^{\text{des}}(t) = \begin{bmatrix} \omega_i^{\text{des}}(t) \\ v_i^{\text{des}}(t) \end{bmatrix}, \quad [\omega_i^{\text{des}}(t)]_\times = R_{sd_i}(t)^\top \dot{R}_{sd_i}(t), \quad v_i^{\text{des}}(t) = R_{sd_i}(t)^\top \dot{p}_{sd_i}(t), \tag{4}$$

where $[\omega]_\times$ denotes the skew-symmetric matrix of ω. Both $\omega_i^{\text{des}}(t)$ and $v_i^{\text{des}}(t)$ are expressed in the desired frame $\{d_i\}$ by construction.

Step 3: Stable Imitation Vector Field. As execution progresses, actual end-effector poses may deviate significantly from the desired trajectory due to tracking errors, disturbances, and model mismatch. To address the **open-loop nature** of waypoints and provide robustness, we construct a vector field that balances **imitation** of demonstrated motions and **stability** for error correction:

$$\mathcal{V}_i^{\text{ref}}(t, T_{sb_i}) = \text{Ad}_{T_{b_i d_i}} \mathcal{V}_i^{\text{des}}(t) + k_{p_i} \mathcal{E}_i, \tag{5}$$

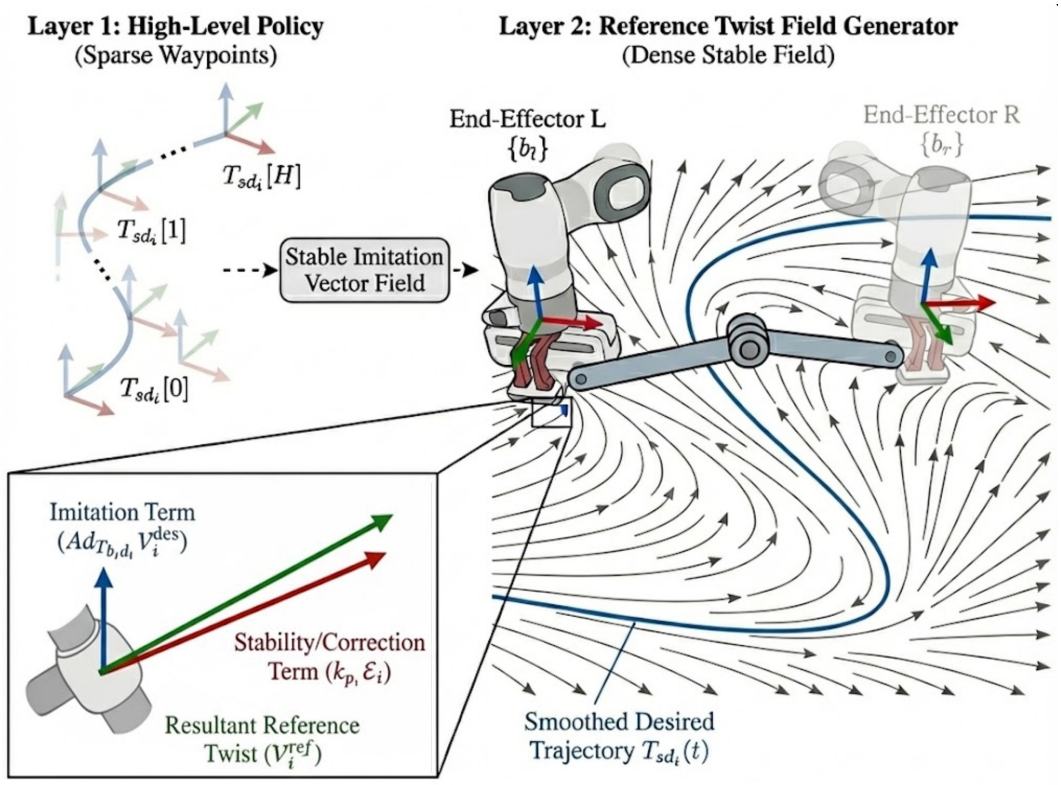


Figure 2: **Reference Twist Field Generator.** The generator transforms sparse high-level waypoints into dense, stable reference twists. The stable imitation vector field combines the desired twist with a pose-error corrective term $k_{p_i}\mathcal{E}_i$, that balances imitation fidelity with robustness to deviations.

where $\mathrm{Ad}_{T_{b_i d_i}}$ is the adjoint transformation mapping the desired twist from frame $\{d_i\}$ to the current body frame $\{b_i\}$. The transformation $T_{b_i d_i} = (T_{sb_i})^{-1} T_{sd_i}$ represents the relative pose from desired to current frame. The pose error term is:

$$\mathcal{E}_i = \begin{pmatrix} \alpha\, e_{R_i} \\ e_{p_i} \end{pmatrix} \in \mathbb{R}^6, \quad e_{R_i} = \log(R_{sb_i}^\top R_{sd_i})^\vee, \quad e_{p_i} = R_{sb_i}^\top (p_{sd_i} - p_{sb_i}), \tag{6}$$

where $e_{R_i} \in \mathbb{R}^3$ is the rotation error and $e_{p_i} \in \mathbb{R}^3$ is the translation error, both expressed in the body frame. Here $\alpha \in \mathbb{R}$ is a characteristic length weighting rotation vs. translation from the riemannian metric(Appendix B), and $k_{p_i} \in \mathbb{R}$ is a proportional gain of corrective field for pose error.

Metric Tensor. The characteristic length α defines a metric tensor $G = \mathrm{diag}(\alpha^2 I_3, I_3)$ inducing an inner product on $\mathfrak{se}(3)$. Physically, this inner product corresponds to the *kinetic energy* of an isotropic rigid body with unit mass and scalar moment of inertia $\alpha^2 I_3$:

$$\langle \mathcal{V}_1, \mathcal{V}_2 \rangle_G = \frac{\alpha^2}{2} \mathrm{tr}([\omega_1]^\top [\omega_2]) + v_1^\top v_2 = \mathcal{V}_1^\top G \mathcal{V}_2, \tag{7}$$

Here, α acts as the radius of gyration, representing the ratio between the body's rotational and translational inertia. Furthermore, this metric induces a left-invariant Riemannian metric on $SE(3)$ with a distinct geometric significance: its geodesics correspond to the *free-body motion* of said isotropic body, thereby providing a dynamically consistent notion of distance on the manifold.

Dual Role of $k_p\mathcal{E}$. The stability term $k_{p_i}\mathcal{E}_i$ serves a crucial dual purpose: it provides corrective feedback for trajectory tracking *and* acts as an elastic force term in the impedance framework. As we show in Section 3.4.3, this formulation **bypasses the problematic SE(3) Jacobian** $J_{\mathcal{E}}$ while maintaining impedance behavior.

3.4 Screw Axes-Decomposed Impedance Control (Layer 4)

This layer addresses **C2** and **C3** by enabling compliant control with independent regulation of bulk and internal motions. Rather than enforcing hard kinematic constraints (which is brittle under model uncertainty), SWIVL adopts a **compliant approach** through impedance control, providing soft regulation of constraint violations while minimizing harmful bulk forces.

The key insight is to *decompose* the twist space into object-centric subspaces aligned with the object's screw axes.

3.4.1 Motion Space Decomposition

In articulated object manipulation, tracking is fundamentally object-centric. To maximally reflect the task semantics embedded in the high-level cognitive planner's trajectory, it is essential to interpret the motion relative to the object structure. Consequently, rather than independently tracking each arm's reference motion, it is advantageous to decompose motions into:

- **Internal motion** $\mathcal{V}_\parallel \in \text{range}(J_i)$: drives joint articulation
- **Bulk motion** $\mathcal{V}_\perp$: drives the object's overall motion through space

Tracking these components separately provides clearer learning signals aligned with the object's kinematic structure.

Let $J_i(\mathbf{q}_{\text{obj}}) \in \mathbb{R}^{6\times k}$ denote the body Jacobian encoding how object joint velocities $\dot{\mathbf{q}}_{\text{obj}}$ manifest as end-effector body twists. The body Jacobian relates to the spatial Jacobian via the adjoint: $J_i = \text{Ad}_{T_{b_i s}}\mathbf{J}_s$, where $T_{b_i s} = (T_{s b_i})^{-1}$.

Using the metric tensor $G = \text{diag}(\alpha^2 I_3, I_3)$ from Eq. (7), we construct G-orthogonal projection operators:

$$P_{i,\parallel} = J_i(J_i^\top G J_i)^{-1} J_i^\top G, \quad P_{i,\perp} = I - P_{i,\parallel}, \tag{8}$$

where $P_{i,\parallel}$ projects onto the internal motion subspace (range of J_i) and $P_{i,\perp}$ projects onto the bulk motion subspace (orthogonal complement). These projectors satisfy:

- G-self-adjointness: $P_{i,\parallel}^\top G = G P_{i,\parallel}$ and $P_{i,\perp}^\top G = G P_{i,\perp}$
- Orthogonality: $P_{i,\parallel} P_{i,\perp} = 0$
- Partition of identity: $P_{i,\parallel} + P_{i,\perp} = I$

Importantly, since α is learned by the policy (Section 3.5), the metric tensor G and hence the projection operators adapt dynamically to task requirements, enabling context-dependent orthogonal decomposition.

Twists decompose as:

$$\mathcal{V}_{i,\parallel}^{\text{ref}} = P_{i,\parallel}\mathcal{V}_i^{\text{ref}}, \quad \mathcal{V}_{i,\parallel} = P_{i,\parallel}\mathcal{V}_i \quad \text{(internal motion)}, \tag{9}$$

$$\mathcal{V}_{i,\perp}^{\text{ref}} = P_{i,\perp}\mathcal{V}_i^{\text{ref}}, \quad \mathcal{V}_{i,\perp} = P_{i,\perp}\mathcal{V}_i \quad \text{(bulk motion)}. \tag{10}$$

3.4.2 Controller Formulation

Given impedance variables $(d_{i,\parallel}, d_{i,\perp}, k_{p_i}, \alpha)$ from the policy, we construct a damping matrix respecting the motion decomposition:

$$K_{d_i} = G(P_{i,\parallel} d_{i,\parallel} + P_{i,\perp} d_{i,\perp}), \tag{11}$$

enabling independent damping for internal motion (via $d_{i,\parallel}$) and bulk motion (via $d_{i,\perp}$). The policy can adaptively regulate compliance based on task requirements and force feedback.

The commanded wrench is:

$$\begin{aligned} \mathcal{F}_{\text{cmd},i} &= K_{d_i}(\mathcal{V}_i^{\text{ref}} - \mathcal{V}_i) + \mu_{b,i} + \gamma_{b,i} \\ &= K_{d_i}(\text{Ad}_{T_{b_i d_i}}\mathcal{V}_i^{\text{des}} - \mathcal{V}_i + k_{p_i}\mathcal{E}_i) + \mu_{b,i} + \gamma_{b,i}, \end{aligned} \tag{12}$$

where $\mathcal{V}_i^{\text{ref}} = \text{Ad}_{T_{b_i d_i}}\mathcal{V}_i^{\text{des}} + k_{p_i}\mathcal{E}_i$ is the reference twist from Layer 2, $\mu_{b,i}$ compensates for Coriolis/centrifugal terms, and $\gamma_{b,i}$ provides gravity compensation (omitted in planar SE(2) settings).

The commanded wrench maps to joint torques via the manipulator Jacobian:

$$\tau_{\text{cmd},i} = J_i(\theta_i)^\top \mathcal{F}_{\text{cmd},i}, \tag{13}$$

where $J_i(\theta_i) \in \mathbb{R}^{6\times n}$ is the geometric Jacobian mapping joint velocities to end-effector twist.

3.4.3 Dual Interpretation of the Controller

The controller admits two complementary interpretations that reveal its power.

Interpretation 1: Twist-Driven SE(3) Impedance. Classical SE(3) impedance control designs a virtual dynamical system:

$$M\dot{\xi} + D\xi + J_{\mathcal{E}}^\top K\mathcal{E} = \mathcal{F}_{\text{ext}}, \tag{14}$$

where $\xi = {}^b\mathcal{V}_d - {}^b\mathcal{V}_b$ is velocity error, M is desired inertia, D is damping, and $J_{\mathcal{E}}^\top K\mathcal{E}$ is the nonlinear stiffness term arising from SE(3) geometry. The corresponding controller takes the form:

$$\mathcal{F}_{\text{cmd}} = \Lambda_b M^{-1}(D\xi + J_{\mathcal{E}}^\top K\mathcal{E}) + \Lambda_b {}^b\dot{\mathcal{V}}_d + \mu_b + \gamma_b + (I - \Lambda_b M^{-1})\mathcal{F}_{\text{ext}}, \tag{15}$$

where Λ_b is the operational space inertia. Under common simplifications $M = \Lambda_b$ and ${}^b\dot{\mathcal{V}}_d = 0$, this reduces to:

$$\mathcal{F}_{\text{cmd}} = D\xi + J_{\mathcal{E}}^\top K\mathcal{E} + \mu_b + \gamma_b. \tag{16}$$

Our controller in Eq. (12) follows this structure. Defining velocity error $\xi = \text{Ad}_{T_{b_i d_i}} \mathcal{V}_i^{\text{des}} - \mathcal{V}_i$:

$$\begin{aligned} \mathcal{F}_{\text{cmd},i} &= K_{d_i}\xi + K_{d_i}k_{p_i}\mathcal{E}_i + \mu_{b,i} + \gamma_{b,i} \\ &\approx D\xi + J_{\mathcal{E}}^\top K\mathcal{E} + \mu_b + \gamma_b, \end{aligned} \tag{17}$$

with correspondence $D \leftrightarrow K_{d_i}$ (learned damping) and $J_{\mathcal{E}}^\top K\mathcal{E} \leftrightarrow K_{d_i}k_{p_i}\mathcal{E}_i$ (stiffness). Critically, our approach **sidesteps the explicit nonlinear Jacobian** $J_{\mathcal{E}}$ by incorporating pose error directly into the reference twist, avoiding geometric complications while maintaining impedance behavior.

Interpretation 2: Task-Semantic Motion Decomposition. Decomposing twists via Eqs. (9)–(10), the control law expands to:

$$\begin{aligned} \mathcal{F}_{\text{cmd},i} &= G(P_{i,\|}d_{i,\|} + P_{i,\perp}d_{i,\perp})(\mathcal{V}_i^{\text{ref}} - \mathcal{V}_i) + \mu_{b,i} + \gamma_{b,i} \\ &= \underbrace{d_{i,\|}G(\mathcal{V}_{i,\|}^{\text{ref}} - \mathcal{V}_{i,\|})}_{\text{internal motion control}} + \underbrace{d_{i,\perp}G(\mathcal{V}_{i,\perp}^{\text{ref}} - \mathcal{V}_{i,\perp})}_{\text{bulk motion control}} + \mu_{b,i} + \gamma_{b,i}. \end{aligned} \tag{18}$$

This decomposition provides two critical properties:

1. **Independent compliance modulation**: The policy can independently adjust $d_{i,\|}$ and $d_{i,\perp}$ for task-specific behavior—high stiffness for bulk motion during transport, high compliance for bulk motion during articulation, or vice versa. It gives implicit constraint satisfaction ensuring compliant motions respect the object's kinematic constraints.
2. **Decoupled power generation**: The feedback wrenches for internal and bulk motions are **reciprocally orthogonal** to opposing motion subspaces:

$$\mathcal{F}_{\text{cmd,fb},i,\|} = d_{i,\|}G(\mathcal{V}_{i,\|}^{\text{ref}} - \mathcal{V}_{i,\|}), \tag{19}$$

$$\mathcal{F}_{\text{cmd,fb},i,\perp} = d_{i,\perp}G(\mathcal{V}_{i,\perp}^{\text{ref}} - \mathcal{V}_{i,\perp}), \tag{20}$$

$$(\mathcal{F}_{\text{cmd,fb},i,\|})^\top(\mathcal{V}_{i,\perp}^{\text{ref}} - \mathcal{V}_{i,\perp}) = 0, \tag{21}$$

$$(\mathcal{F}_{\text{cmd,fb},i,\perp})^\top(\mathcal{V}_{i,\|}^{\text{ref}} - \mathcal{V}_{i,\|}) = 0. \tag{22}$$

   This orthogonality follows from $P_{i,\|}^\top G P_{i,\perp} = 0$, ensuring control actions for each motion type do not interfere.

Together, these interpretations demonstrate that our controller achieves geometrically consistent SE(3) impedance behavior while enabling explicit, learning-based modulation of task-semantic motion components—a capability that would be intractable to design analytically.

3.5 Wrench-Adaptive Impedance Learning (Layer 3)

This layer addresses **C3** and **C4** by learning adaptive impedance modulation. The RL policy $\pi_\theta : \mathcal{O} \to \mathcal{A}$ modulates impedance variables to enable physically feasible motions while accounting for object constraints and inter-arm force interactions. By explicitly conditioning on object geometry (screw axes) and wrench feedback, the policy learns to independently modulate compliance for bulk versus internal motions and mitigating unnecessary bulk forces.

3.5.1 Observation and Action Spaces

Observation Space $\mathcal{O}$. The policy receives:

1. **Reference twists**: $\{\mathcal{V}_l^{\text{ref}}, \mathcal{V}_r^{\text{ref}}\} \in \mathfrak{se}(3) \times \mathfrak{se}(3)$
   Reference motions computed by Reference Twist Field Generator (Layer 2) with desired twist and pose error correction.
2. **Object kinematic structure**:
   For a k-DoF articulated object,
   - **Inter-link screw axes**: $\{\mathcal{A}_2, \ldots, \mathcal{A}_{k-1}\}$ where $\mathcal{A}_i \in \mathfrak{se}(3)$ is the screw axis of joint i expressed in the $(i-1)$-th link frame, describing the relative motion from link $(i-1)$ to link i.
   - **Body Screw of $1^{th}, k^{th}$ joint**: $\{\mathcal{B}_l, \mathcal{B}_r\} \in \mathfrak{se}(3) \times \mathfrak{se}(3)$
     Represents body-frame screw axes of $1^{th}, k^{th}$ joint at each corresponding end-effector.

   For **1-DoF objects**, the kinematic structure reduces to the body-frame screw axes $\{\mathcal{B}_l, \mathcal{B}_r\} \in \mathfrak{se}(3) \times \mathfrak{se}(3)$. This simplified representation is sufficient for our experimental validation in SE(2) with single-joint articulated objects.
3. **Wrench feedback**: $\{\mathcal{F}_l, \mathcal{F}_r\} \in \mathfrak{se}(3)^* \times \mathfrak{se}(3)^*$
   6-dimensional wrench measurements (3D moment + 3D force) from wrist-mounted F/T sensors.
4. **Proprioception**: $\{T_{sb_l}, T_{sb_r}, \mathcal{V}_l, \mathcal{V}_r\} \in \mathrm{SE}(3) \times \mathrm{SE}(3) \times \mathfrak{se}(3) \times \mathfrak{se}(3)$
   Task-space states including end-effector poses and body twists.

Action Space $\mathcal{A}$. We propose an impedance variable action space that structurally enables object-aware compliance modulation:

$$a_t = (d_{l,\|}, d_{r,\|}, d_{l,\perp}, d_{r,\perp}, k_{p_l}, k_{p_r}, \alpha) \in \mathbb{R}^7, \tag{23}$$

where all variables are positive scalars:

- $d_{i,\|}$: Damping coefficient for internal motion.
- $d_{i,\perp}$: Damping coefficient for bulk motion.
- k_{p_i}: Stiffness coefficient for the stability term in the reference twist.
- α: Learnable characteristic length that adaptively modulates the metric tensor.

These variables parameterize the low-level controller (Section 3.4.2), enabling adaptive, context-dependent impedance modulation. Note that, by learning α, the policy discovers task-appropriate metric structures for orthogonal decomposition.

3.5.2 Wrench Decomposition and Force Regulation

To resolve the **bulk motion ambiguity** (**C4**), we decompose measured wrenches into internal and bulk components. If high-level planners generated kinematically consistent commands, bulk motion components would agree. However, planners lack constraint awareness, resulting in mismatched bulk motions that manifest as inter-arm forces.

By twist-wrench duality, projecting wrenches with transposed twist projectors preserves orthogonality under the reciprocal product (virtual power). We seek wrench components $\mathcal{F}_{i,\|}$ and $\mathcal{F}_{i,\perp}$

satisfying:

$$\langle \mathcal{F}_{i,\|}, \mathcal{V} \rangle = \mathcal{F}_{i,\|}^\top \mathcal{V} = 0 \quad \forall \mathcal{V} \in \text{range}(P_{i,\perp}), \tag{24}$$

$$\langle \mathcal{F}_{i,\perp}, \mathcal{V} \rangle = \mathcal{F}_{i,\perp}^\top \mathcal{V} = 0 \quad \forall \mathcal{V} \in \text{range}(P_{i,\|}). \tag{25}$$

This is naturally satisfied by:

$$\mathcal{F}_{i,\|} = P_{i,\|}^\top \mathcal{F}_i \quad \text{(internal)}, \qquad \mathcal{F}_{i,\perp} = P_{i,\perp}^\top \mathcal{F}_i \quad \text{(bulk)}. \tag{26}$$

To verify orthogonality: for any $\mathcal{V} \in \text{range}(P_{i,\perp})$, we have $\mathcal{V} = P_{i,\perp}\mathcal{V}'$, and:

$$\mathcal{F}_{i,\|}^\top \mathcal{V} = (P_{i,\|}^\top \mathcal{F}_i)^\top (P_{i,\perp}\mathcal{V}') = \mathcal{F}_i^\top P_{i,\|} P_{i,\perp} \mathcal{V}' = 0, \tag{27}$$

where the last equality follows from $P_{i,\|}P_{i,\perp} = 0$.

The parallel component $\mathcal{F}_{i,\|}$ represents **internal wrench** performing work along the object's internal DoF. The orthogonal component $\mathcal{F}_{i,\perp}$ represents **internal wrench** that:

- Does not contribute to object's internal joint motion (zero virtual power along range$(P_{i,\|})$)
- Arises from coordination errors and high-level commands that do not match between the two arms.
- May increases contact stress and grasping instability with hardware damage.

Minimizing $\|\mathcal{F}_{i,\perp}\|^2$ implicitly resolves the bulk motion ambiguity: the policy learns to track references in a manner achieving coordinated bulk motion while maintaining compliant internal motion.

3.5.3 Reward Design

The reward balances four objectives for stable, force-compliant manipulation:

$$r_t = r_{\text{track}} + r_{\text{safety}} + r_{\text{reg}} + r_{\text{term}}. \tag{28}$$

Motion Tracking (r_{track}). Encourages reference following with the learned G-metric:

$$r_{\text{track}} = -w_{\text{track}} \sum_{i \in \{l,r\}} \|\mathcal{V}_i - \mathcal{V}_i^{\text{ref}}\|_G^2. \tag{29}$$

Using the G-metric ensures tracking error is measured consistently with the impedance framework, with adaptive weighting via the learned α.

Safe Inter-arm Force Interaction (r_{safety}). We employ an exponential safety reward that provides a positive 'alive bonus' when fighting forces are low:

$$r_{\text{safety}} = w_{\text{safety}} \exp\left(-\kappa \sum_{i \in \{l,r\}} \|\mathcal{F}_{i,\perp}\|_{G^{-1}}^2 \right), \tag{30}$$

where $\kappa > 0$ is a decay rate and $\|\cdot\|_{G^{-1}}$ denotes the dual metric norm. The dual metric $G^{-1} = \text{diag}(1/\alpha^2, I_3)$ is mathematically induced from the twist metric G and ensures dimensional consistency between moment and force components. This formulation has two key properties: (i) when $\|\mathcal{F}_{i,\perp}\| \approx 0$, the policy receives $r_{\text{safety}} \approx w_{\text{safety}}$, incentivizing safe operation; (ii) as fighting forces increase, the reward decays smoothly toward zero. Unlike quadratic penalties (all negative rewards), this exponential form prevents the pathological behavior where agents learn to terminate early to avoid accumulating negative rewards.

Motion Smoothness (r_{reg}). Regularizes twist acceleration:

$$r_{\text{reg}} = -w_{\text{reg}} \sum_{i \in \{l,r\}} \|\dot{\mathcal{V}}_i\|^2, \tag{31}$$

reducing jerkiness for motion smoothness.

Termination Penalty (r_{term}). Applied upon failure termination:

$$r_{\text{term}} = \begin{cases} -w_{\text{term}} & \text{if failure termination} \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

This explicit penalty discourages the agent from learning behaviors that intentionally trigger termination conditions.

Termination Conditions. Episodes terminate (with r_{term} applied) under two failure conditions:

(1) Grasp Drift. If grasp pose drift exceeds a threshold:

$$\exists i \in \{l, r\} : \left\| \left[\log \left((T^{\text{init}}_{\text{grip},i})^{-1} T_{\text{grip},i} \right) \right]^{\vee} \right\|_G > d_{\max}, \tag{33}$$

where $T^{\text{init}}_{\text{grip},i}$ and $T_{\text{grip},i}$ are initial and current grasp poses.

(2) Wrench Limit. If external wrenches exceed safe operating limits:

$$\exists i \in \{l, r\} : \|\mathcal{F}_i\|_{G^{-1}} > \mathcal{F}_{\max}, \tag{34}$$

where $\|\mathcal{F}\|^2_{G^{-1}} = \mathcal{F}^\top G^{-1} \mathcal{F}$ uses the dual metric G^{-1} to weight moment and force components consistently with the twist metric G. Excessive wrenches risk hardware damage, grasp slippage, and object deformation.

3.5.4 Policy Architecture and Training

The policy $\pi_\theta : \mathcal{O} \to \mathcal{A}$ uses a multi-stream architecture with Feature-wise Linear Modulation (FiLM) to inject object geometry into feature processing. Separate encoders process reference twists, wrenches, and proprioception, with FiLM layers modulating features based on screw axes $\{\mathcal{B}_l, \mathcal{B}_r\}$, enabling dynamic adaptation across joint types.

We train with **Proximal Policy Optimization (PPO)** Schulman et al. [2017]:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right], \tag{35}$$

where $r_t(\theta) = \pi_\theta(a_t|o_t)/\pi_{\theta_{\text{old}}}(a_t|o_t)$ and advantages are computed via Generalized Advantage Estimation (GAE) Schulman et al. [2015]:

$$\hat{A}_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}, \quad \delta_t = r_t + \gamma V_\phi(o_{t+1}) - V_\phi(o_t), \tag{36}$$

where $V_\phi : \mathcal{O} \to \mathbb{R}$ is the value function with architecture identical to the policy encoder. See Appendix D for detailed architecture specifications.

4 Experiments

We evaluate SWIVL on bimanual manipulation of articulated objects in an SE(2) planar benchmark. Our experiments aim to verify whether SWIVL improves task success and reduces fighting forces, comparing its performance against high stiffness position control and linearized impedance control.

4.1 Experimental Setup

SE(2) Benchmark Rationale. While our method (Section 3) is formulated for SE(3) with k-DoF articulated objects, we validate in SE(2) with 1-DoF objects. This deliberate simplification enables rigorous evaluation while preserving the essential challenges: **(i)** the fundamental phenomena—force coupling, constraint satisfaction, compliant coordination—manifest identically in planar and spatial settings; **(ii)** all architectural components (projection operators $P_{i,\|}, P_{i,\perp}$, metric tensor $G(\alpha)$, impedance modulation $d_\|, d_\perp$) remain fully exercised; and **(iii)** the mathematical structure (Lie group, screw theory, twist-wrench duality) reduces consistently from SE(3). Extension to SE(3) requires only scaling observation/action dimensions. See Appendix C for complete SE(2) instantiation.

Algorithm 1 SWIVL Training

Require: High-Level Policy π_{HL}, object set $\mathcal{O}_{\text{obj}}$
Initialize policy parameters θ, value function parameters ϕ
for episode = 1 to N_{episodes} **do**
  Sample object $o \sim \mathcal{O}_{\text{obj}}$; initialize environment
  for $t = 1$ to H **do**
    // Layer 1: High-Level Policy
    if $t \mod f_{HL}^{-1} = 0$ **then**
      Generate action chunk $\{T_{sd_i}[\tau]\}_{\tau=0}^{H_{\text{chunk}}} \leftarrow \pi_{HL}$
    end if
    // Layer 2: Reference Twist Field Generator
    Interpolate action chunk $\rightarrow$ dense trajectory $T_{sd_i}(t)$
    Compute desired body twists $\mathcal{V}_i^{\text{des}}(t)$ via Eq. (4)
    Apply stable vector field $\rightarrow$ reference twists $\mathcal{V}_i^{\text{ref}}$ via Eq. (5)
    Decompose $\mathcal{V}_i^{\text{ref}} \rightarrow$ internal $\mathcal{V}_{i,\|}^{\text{ref}}$ and bulk $\mathcal{V}_{i,\perp}^{\text{ref}}$
    // Layer 3: Impedance Modulation Policy
    Observe $o_t = (\mathcal{V}_i^{\text{ref}}, \mathcal{B}_i, \mathcal{F}_i, T_{sb_i}, \mathcal{V}_i)$
    Sample $(d_{i,\|}, d_{i,\perp}, k_{p_i}, \alpha) \sim \pi_\theta(\cdot|o_t)$
    // Layer 4: Screw-Decomposed Controller
    Compute commanded wrench $\mathcal{F}_{\text{cmd},i}$ via Eq. (12)
    Execute joint torque $\tau_{\text{cmd},i} = J_i(\theta_i)^\top \mathcal{F}_{\text{cmd},i}$
    // Collect Experience
    Observe o_{t+1}, compute reward r_t via Eq. (28)
    Store transition (o_t, a_t, r_t, o_{t+1})
  end for
  // PPO Update
  Compute advantages via GAE (Eq. (36))
  for epoch = 1 to K **do**
    Sample mini-batches; update θ via Eq. (35); update ϕ via MSE loss
  end for
end for

Environment. We introduce **BiarT** (Bimanual Articulated manipulation), a Pymunk-based SE(2) simulation environment for bimanual manipulation of articulated objects. Just as **PushT** Chi et al. [2025] serves as a minimal benchmark for single-arm planar manipulation, BiarT provides a lightweight testbed for studying the unique challenges of bimanual coordination—force coupling, constraint satisfaction, and compliant control—without the complexity of full 3D physics. The environment features a 512×512 pixel planar workspace with dual 3-DoF end-effectors under direct body wrench control $\mathcal{F}_i = [m_z, f_x, f_y]^\top$. Each end-effector provides 3-axis F/T sensing. The hierarchical architecture combines high-level planning (10 Hz) with low-level control (100 Hz).

Tasks and Objects. We evaluate on **2 articulated objects** spanning two joint types (Figure 3):

- **Revolute**: Angular articulation—rotation about a pivot point. One arm drives rotation while the other must comply.
- **Prismatic**: Linear articulation—sliding along an axis. One arm drives extension/retraction while the other accommodates.

Each object satisfies the SE(2) holonomic constraint ${}^s\mathcal{V}_l - {}^s\mathcal{V}_r = \mathcal{S}\dot{q}_{obj}$ with constant body-frame screw axes $\mathcal{B}_l, \mathcal{B}_r \in \mathbb{R}^3$. Tasks require manipulating objects from randomized initial configurations to goal configurations. **Success criteria**: position error <10 pixels, orientation error <5°, joint error <5° or 5 pixels, with maintained grasp. Each configuration is tested over **10 trials**.

High-Level Planner. All methods share a common high-level planner: a **Flow Matching Policy** trained via behavior cloning on expert demonstrations. The policy outputs action chunks—sequences of desired end-effector poses $\{T_{sd_i}[\tau]\}_{\tau=0}^{H}$—at 10 Hz. This architecture represents state-of-the-art imitation learning for manipulation, providing temporally consistent motion intentions

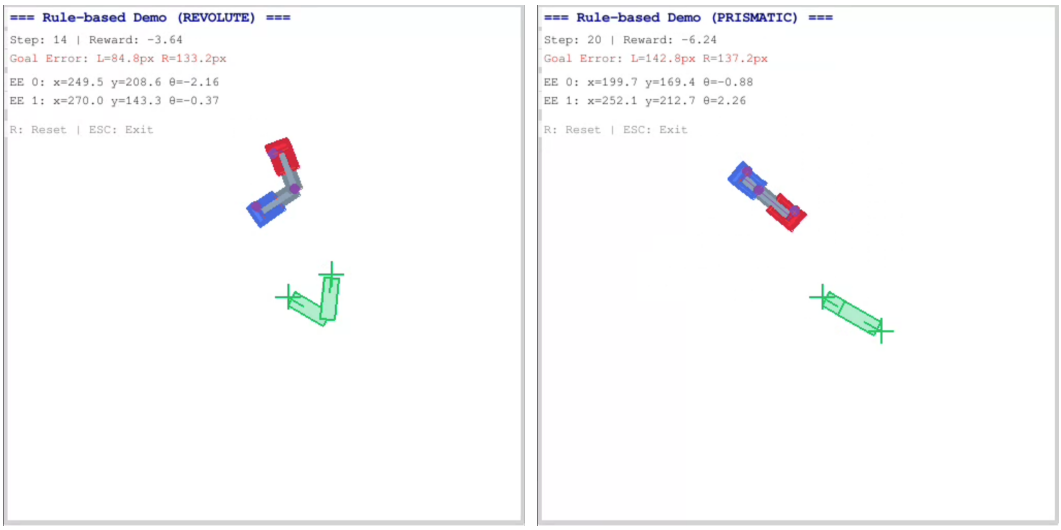


Figure 3: **Benchmark objects.** Two SE(2) articulated objects spanning two joint types: revolute (angular articulation) and prismatic (linear articulation). Red and Blue end-effectors are floating base parallel grippers.

without explicit reasoning about contact dynamics or inter-arm force coordination. By fixing the high-level planner across all methods, we isolate the contribution of the low-level control strategy.

Methods Under Comparison. We compare three low-level control strategies, all receiving identical action chunks from the Flow Matching Policy:

- **Position Control (Pos-Ctrl)**: The action chunks are tracked via high-stiffness position control. This represents the standard deployment of imitation learning policies, where learned poses are directly executed without force awareness. The controller applies:
$$\mathcal{F}_{cmd,i} = K_p(p_{d_i} - p_i) + K_d(\dot{p}_{d_i} - \dot{p}_i) \tag{37}$$
with high gains K_p, K_d to minimize tracking error, prioritizing trajectory fidelity over compliance.
- **Impedance Control (Imp-Ctrl)**: The action chunks are tracked via linearized SE(2) impedance control with linearized approximations. This baseline employs the standard mass-spring-damper formulation:
$$\mathcal{F}_{cmd,i} = D\xi_i + K\mathcal{E}_i + \mu_{b,i} \tag{38}$$
where ξ_i is the twist error, $\mathcal{E}_i$ is the pose error, and D, K are fixed diagonal damping and stiffness matrices. This approach assumes small orientation errors ($J_l^{-1}(e_R) \approx I_3$) and uses isotropic impedance parameters without object-aware decomposition.
- **SWIVL (Ours)**: The action chunks are processed through the full SWIVL framework: (i) the Reference Twist Field Generator converts sparse waypoints into dense, stable reference twists with pose-error correction, (ii) the RL policy modulates impedance variables $(d_{i,\parallel}, d_{i,\perp}, k_{p_i}, \alpha)$ conditioned on screw axes and wrench feedback, and (iii) the Screw-Decomposed Impedance Controller executes compliant control with independent regulation of bulk and internal motions.

Implementation. For SWIVL, the policy observes reference twists $\mathcal{V}_i^{\text{ref}}$, screw axes $\mathcal{B}_i$, wrenches $\mathcal{F}_i$, and proprioception ($\mathbb{R}^{30}$ total), and outputs impedance variables $(d_{l,\parallel}, d_{r,\parallel}, d_{l,\perp}, d_{r,\perp}, k_{pl}, k_{pr}, \alpha) \in \mathbb{R}^7$. Training uses PPO with the reward from Eq. (28). For fair comparison, Imp-Ctrl uses the same control frequency (100 Hz) and receives identical trajectory inputs. Full architecture and hyperparameters are in Appendix D.

Metrics.

- **Success Rate** (%): Task completion within error thresholds and time limit

Table 1: **Quantitative comparison.** Performance across two joint types (10 trials each). All methods use the same Flow Matching Policy as high-level planner. Bold indicates best. Timeout failures (not shown) are implicitly 100 − Success − Wrench − Grasp.

Method	Success (%)↑	Wrench Fail (%)↓	Grasp Fail (%)↓
Revolute Joint			
Pos-Ctrl	10	70	0
Imp-Ctrl	10	20	30
SWIVL (Ours)	**40**	**0**	**10**
Prismatic Joint			
Pos-Ctrl	30	40	0
Imp-Ctrl	60	0	20
SWIVL (Ours)	**80**	**0**	**0**
Average (All Objects)			
Pos-Ctrl	20	55	0
Imp-Ctrl	35	10	25
SWIVL (Ours)	**60**	**0**	**5**

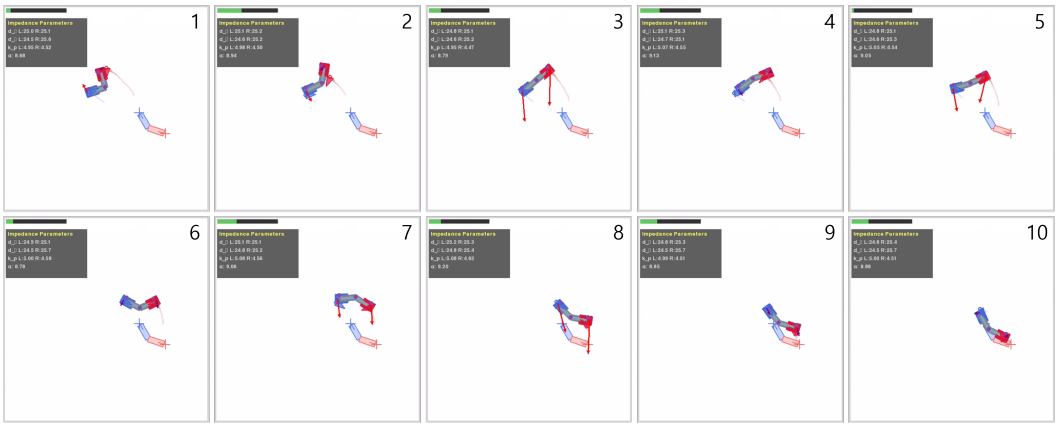


Figure 4: **SWIVL inference sequence.** Time-ordered frames showing successful bimanual manipulation of an articulated object. The impedance parameters panel (left) displays learned values: damping coefficients $d_{\parallel}, d_{\perp}$ for internal and bulk motions, stiffness k_p, and characteristic length α. Red arrows indicate wrench feedback from F/T sensors. The policy adaptively modulates compliance to achieve coordinated manipulation while maintaining low fighting forces.

- **Wrench Limit Failure** (%): Episodes terminated due to exceeding safe wrench thresholds
- **Grasp Drift Failure** (%): Episodes terminated due to grasp instability causing object loss
- **Timeout Failure** (%): Episodes where task was not completed within time limit (implicitly computed as 100 − Success − Wrench Limit − Grasp Drift); these represent performance failures without safety violations

4.2 Results

Table 1 presents quantitative results across all joint types. We observe consistent patterns across the two object categories. Figure 4 illustrates a successful manipulation sequence using the full SWIVL framework, demonstrating coordinated bimanual control with learned impedance modulation.

Success Rate. SWIVL achieves the highest success rates across both joint types, with an average improvement of 40% over Pos-Ctrl and 25% over Imp-Ctrl. The performance gap is particularly pronounced for the revolute joint, where rotational constraints impose stricter coordination requirements that constraint-unaware controllers cannot satisfy.

Wrench Limit Failures. Pos-Ctrl exhibits the highest wrench limit failure rates (55% average) because high-stiffness tracking of kinematically inconsistent references directly translates to excessive inter-arm stress that exceeds safe operating thresholds. This is particularly severe for the revolute joint (70%), where rotational constraints amplify force coupling. Imp-Ctrl reduces these failures to 10% through passive compliance. SWIVL achieves zero wrench limit failures by explicitly decomposing wrenches into productive (along constraint) and non-productive (orthogonal) components, then learning to suppress the latter through impedance modulation.

Grasp Drift Failures. Interestingly, Imp-Ctrl exhibits the highest grasp drift failures (25% average) despite having lower wrench failures than Pos-Ctrl. This reveals a fundamental trade-off: isotropic compliance reduces peak forces but introduces uncontrolled motion that destabilizes the grasp over time. Pos-Ctrl shows zero grasp drift because its rigid tracking prevents gradual slip—though this comes at the cost of wrench limit violations. SWIVL achieves only 5% grasp drift by using screw-decomposed control that selectively applies compliance along constraint directions while maintaining stability orthogonal to them.

Timeout Failures. The remaining failures (not shown in Table 1) are timeout failures—episodes where the task was not completed within the time limit without any safety violations. Notably, SWIVL has higher timeout rates (35% average) than Pos-Ctrl (25%) and Imp-Ctrl (30%), which reflects a deliberate trade-off: by prioritizing safety through learned compliance, SWIVL may occasionally be more conservative in its motions. However, this "safe but slow" behavior is preferable in real-world deployment where safety violations can cause hardware damage or task failure.

4.3 Analysis

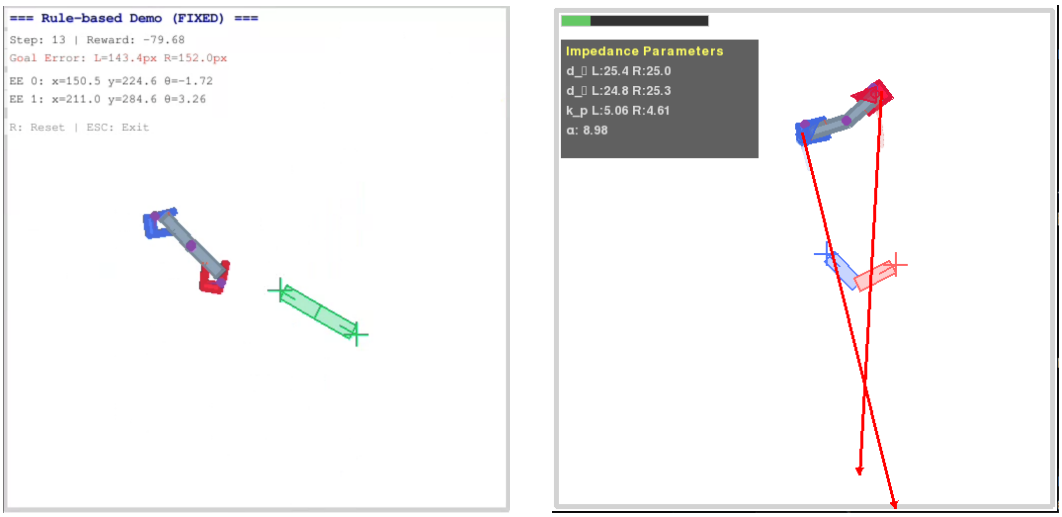
Why Position Control Fails. Position control's fundamental limitation is its inability to accommodate kinematic constraints. When the high-level Flow Matching Policy outputs action chunks, these represent motion *intentions* learned from demonstrations—they do not guarantee kinematic consistency between the two arms. Under high-stiffness control, even small inconsistencies (<1% position error) accumulate into substantial fighting forces because the controller treats the reference as a hard constraint to be achieved regardless of physical consequences. This is particularly problematic for articulated objects where the holonomic constraint ${}^{s}\mathcal{V}_l - {}^{s}\mathcal{V}_r = \mathcal{S}\dot{q}_{obj}$ must be satisfied continuously. Figure 5 illustrates two representative failure modes: (a) grasp drift caused by excessive fighting forces that destabilize the grasp (Figure 5a), and (b) wrench limit violations where uncontrolled forces exceed safe operating thresholds (Figure 5b).

Why Classical Impedance Control Is Insufficient. Classical impedance control introduces compliance but suffers from two key limitations in our setting:

1. **Linearization errors**: The approximation $J_l^{-1}(e_R) \approx I_3$ assumes small orientation errors, which is violated during articulated manipulation where constraint mismatches can cause large trajectory deviations.
2. **Lack of object awareness**: Fixed isotropic impedance parameters cannot distinguish between motion along the object's kinematic constraint (which should be compliant) and motion orthogonal to it (which may require stiffness for stability). This one-size-fits-all approach is fundamentally mismatched to articulated object manipulation.

SWIVL's Advantages. SWIVL addresses these limitations through three mechanisms:

1. **Twist-driven formulation**: By incorporating pose errors into reference twists rather than computing elastic wrenches explicitly, SWIVL bypasses the nonlinear pose-error Jacobian J_E.
2. **Screw-decomposed control**: The projection operators $P_{i,\parallel}, P_{i,\perp}$ partition the twist space into object-centric subspaces, enabling independent compliance for internal versus bulk motions.
3. **Learned impedance modulation**: The RL policy discovers task-appropriate impedance strategies conditioned on real-time wrench feedback and object geometry, adapting compliance dynamically rather than relying on fixed parameters.



(a) Grasp drift failure (b) Wrench limit exceeded

Figure 5: **Failure modes in experiments.** (a) *Grasp drift* (b) *Wrench limit exceeded*

5 Discussion

5.1 Key Findings

Our SE(2) experiments confirm that **explicit encoding of geometric constraints and wrench feedback** is essential for robust bimanual manipulation of articulated objects. Three key findings emerge from our comparison of position control, classical impedance control, and SWIVL:

Position Control Is Fundamentally Limited. High-stiffness position control treats each arm's reference trajectory as an independent hard constraint, ignoring the physical coupling between arms grasping a shared object. When the high-level Flow Matching Policy outputs kinematically inconsistent commands—inevitable for any learning-based planner without explicit constraint modeling—rigid tracking generates excessive fighting forces that destabilize grasps and cause task failures. This limitation is intrinsic to the control paradigm, not the quality of demonstrations.

Classical Impedance Control Is Insufficient. While impedance control introduces compliance that partially absorbs coordination errors, its isotropic and fixed parameters cannot distinguish between motion along the object's kinematic constraint (which should be compliant) and motion orthogonal to it (which may require stiffness). The linearized SE(3) formulation further breaks down under large trajectory deviations caused by constraint violations, leading to unpredictable behavior.

Object-Aware Decomposition Enables Physical Intelligence. SWIVL's success stems from explicitly representing the object's kinematic structure through screw-axis-based projection operators. By decomposing motions and wrenches into bulk (transport) and internal (articulation) components, the learned policy can modulate compliance independently for each subspace. This geometric grounding, combined with real-time wrench feedback, enables the policy to discover force-compliant strategies that position control cannot express and classical impedance control cannot adapt to.

5.2 Limitations and Future Work

Current Limitations.

- **SE(2) planar setting**: Our evaluation is restricted to planar manipulation with 1-DoF articulated objects. Real-world tasks require full SE(3) workspace with multi-DoF constraints.
- **Simulation only**: Sim-to-real transfer remains to be validated on physical dual-arm platforms.

- **Known object kinematics**: We assume screw axes $\mathcal{B}_l, \mathcal{B}_r$ are provided a priori, requiring either CAD models or prior estimation.
- **Fixed high-level planner**: Experiments use a single Flow Matching Policy; generalization across diverse planners (diffusion policies, ACT, teleoperation) is not yet validated.
- **Pre-grasped objects**: Grasping acquisition and regrasping strategies are not addressed.

Future Directions.

- **SE(3) extension and real-robot deployment**: The mathematical framework naturally extends to SE(3) by scaling observation/action dimensions. Validation on dual-arm platforms (e.g., Franka Panda, UR5e) with domain randomization for sim-to-real transfer is a priority.
- **Ablation studies**: Systematic analysis of each architectural component—stable vector fields, screw-decomposed control, FiLM conditioning, wrench feedback—would isolate their individual contributions.
- **Cross-planner generalization**: Evaluating SWIVL with diverse high-level planners would validate the claim that our low-level controller is planner-agnostic.
- **Novel object transfer**: Testing on unseen object geometries and mass distributions would assess the generalization capability of FiLM-based object conditioning.
- **Online constraint estimation**: Integrating screw-axis estimation (e.g., from Screw-Splatting) would eliminate the requirement for known object models.
- **Multi-DoF articulation**: Extending to k-DoF objects with Jacobian $J_i \in \mathbb{R}^{6\times k}$ would broaden applicability to complex articulated structures.

5.3 Broader Impact

SWIVL addresses a critical gap in deploying learning-based manipulation policies to contact-rich bimanual tasks. By providing a principled low-level control layer that operates beneath arbitrary high-level planners, our framework enables safer deployment of cognitive policies (including VLAs and foundation models) that lack explicit physical reasoning. The explicit minimization of fighting forces reduces contact stress, improving both task success and safety in human-robot collaboration scenarios. Potential applications include bimanual assembly in manufacturing, surgical assistance, and cooperative manipulation in service robotics.

6 Conclusion

We introduced **SWIVL** (Screw-Wrench informed Impedance Variable Learning), a hierarchical control framework that bridges the gap between high-level cognitive planning and low-level physically grounded execution for bimanual manipulation of articulated objects. SWIVL addresses the fundamental limitation of existing learning-based manipulation approaches: their reliance on high-stiffness position control without explicit reasoning about kinematic constraints or inter-arm force interactions.

Our framework contributes three key technical innovations. First, twist-driven impedance control via stable imitation vector fields bypasses the nonlinear pose-error Jacobian inherent in SE(3) impedance formulations by incorporating pose errors directly into reference twists, enabling geometrically consistent compliance under large trajectory deviations. Second, screw axes-based decomposition partitions the twist and wrench spaces into object-centric internal and bulk motion subspaces, allowing independent compliance modulation aligned with the articulated object's kinematic structure. Third, wrench-adaptive impedance variable learning via reinforcement learning enables the policy to discover task-appropriate impedance strategies conditioned on real-time force feedback and object geometry, suppressing harmful fighting forces while maintaining compliant trajectory tracking.

Our SE(2) experiments validate the core hypothesis: explicit encoding of geometric constraints and wrench feedback is essential for robust bimanual manipulation. SWIVL consistently outperforms position control and classical impedance control baselines in success rate while significantly reducing inter-arm fighting forces across revolute and prismatic joint types. These results demonstrate

that object-aware motion decomposition and learned impedance modulation provide the physical intelligence that cognitive planners lack.

By operating as a planner-agnostic low-level control layer, SWIVL enables safer deployment of vision-language-action models and other cognitive policies for contact-rich bimanual tasks. The framework's principled integration of screw theory, impedance control, and reinforcement learning offers a foundation for extending physical intelligence to more complex articulated manipulation scenarios in SE(3).

References

OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

Lars Ankile, Anthony Simeonov, Idan Shenfeld, Marcel Torne, and Pulkit Agrawal. From imitation to refinement–residual rl for precise assembly. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 01–08. IEEE, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

Tairan He, Wenli Xiao, Toru Lin, Zhengyi Luo, Zhenjia Xu, Zhenyu Jiang, Jan Kautz, Changliu Liu, Guanya Shi, Xiaolong Wang, et al. Hover: Versatile neural whole-body controller for humanoid robots. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9989–9996. IEEE, 2025.

Neville Hogan. Impedance control: An approach to manipulation: Part i—theory. *Journal of Dynamic Systems, Measurement, and Control*, 107(1):1–7, 1985.

Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6023–6029. IEEE, 2019.

Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

Michelle A Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8943–8950. IEEE, 2019.

Kevin M Lynch and Frank C Park. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

Marc H Raibert and John J Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103(2):126–133, 1981.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Markku Suomalainen, Yiannis Karayiannidis, and Ville Kyrki. A survey of robot manipulation in contact. *Robotics and Autonomous Systems*, 156:104224, 2022.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023. Action Chunking with Transformers (ACT).

A Notation and Mathematical Preliminaries

This appendix establishes the mathematical notation used throughout the paper, following the modern robotics framework Lynch and Park [2017] by Frank C. Park. We adopt Lie group formalism for SE(3) and SE(2), which provides a geometric foundation for manipulation.

A.1 Coordinate Frames and Basic Notation

Reference Frames:

- $\{s\}$: Spatial (world) frame
- $\{l\}, \{r\}$: Left and right end-effector body frames
- $\{o\}$: Object body frame

Twist Notation:

- $\mathcal{V}_a$: Twist of frame $\{a\}$ in its own body frame
- ${}^b\mathcal{V}_a$: Twist of frame $\{a\}$ expressed in frame $\{b\}$

A.2 SE(3) and SE(2) Configuration Spaces

Special Euclidean Groups:

- $SE(3) = \left\{ T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} : R \in SO(3), p \in \mathbb{R}^3 \right\}$: Rigid body transformations in 3D
- $SE(2) = \left\{ T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} : R \in SO(2), p \in \mathbb{R}^2 \right\}$: Planar rigid transformations
- T_{ab}: Transformation from frame $\{b\}$ to frame $\{a\}$

A.3 Twists and Wrenches

Twist (Spatial Velocity): A twist $\mathcal{V}$ represents the instantaneous velocity of a rigid body, combining angular and linear components:

- **SE(3)**: $\mathcal{V} = \begin{bmatrix} \omega \\ v \end{bmatrix} \in \mathbb{R}^6$ where $\omega \in \mathbb{R}^3$ is angular velocity and $v \in \mathbb{R}^3$ is linear velocity
- **SE(2)**: $\mathcal{V} = \begin{bmatrix} \omega_z \\ v_x \\ v_y \end{bmatrix} \in \mathbb{R}^3$ where ω_z is angular velocity about z-axis and (v_x, v_y) are planar linear velocities
- **Body twist** $\mathcal{V}_a$: Twist expressed in the moving body frame $\{a\}$
- **Spatial twist** $\mathcal{V}_s$: Twist expressed in the spatial frame $\{s\}$

Wrench (Generalized Force): A wrench $\mathcal{F}$ represents the generalized force acting on a rigid body, combining moment and force:

- **SE(3)**: $\mathcal{F} = \begin{bmatrix} m \\ f \end{bmatrix} \in \mathbb{R}^6$ where $m \in \mathbb{R}^3$ is moment (torque) and $f \in \mathbb{R}^3$ is force
- **SE(2)**: $\mathcal{F} = \begin{bmatrix} m_z \\ f_x \\ f_y \end{bmatrix} \in \mathbb{R}^3$ where m_z is moment about z-axis and (f_x, f_y) are planar forces
- Wrenches naturally pair with twists via power: $P = \mathcal{F}^T \mathcal{V} = m^T \omega + f^T v$

Adjoint Transformation: Transforms twists between coordinate frames:

$$^a\mathcal{V}_c = [Ad_{T_{ab}}]^b\mathcal{V}_c$$

where the adjoint matrix is:

- **SE(3)**: $[Ad_T] = \begin{bmatrix} R & 0 \\ [p]_\times R & R \end{bmatrix} \in \mathbb{R}^{6\times 6}$ for $T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$
- **SE(2)**: $[Ad_T] = \begin{bmatrix} 1 & 0 & 0 \\ y & \cos\theta & -\sin\theta \\ -x & \sin\theta & \cos\theta \end{bmatrix} \in \mathbb{R}^{3\times 3}$ for pose (x, y, θ)

Twists transform via the adjoint: $^a\mathcal{V}_c = [Ad_{T_{ab}}]^b\mathcal{V}_c$. Wrenches transform via the dual adjoint: $^a\mathcal{F}_c = [Ad_{T_{ab}^{-1}}]^{Tb}\mathcal{F}_c$.

A.4 Screw Theory

Screw Axis: A screw $\mathcal{S}$ describes the instantaneous motion axis of a rigid body. For a unit twist $\mathcal{V}$ (i.e., $\|\omega\| = 1$ or $\omega = 0$), the screw is the twist itself.

- **SE(3) Screw**: $\mathcal{S} = \begin{bmatrix} \omega \\ v \end{bmatrix} \in \mathbb{R}^6$
  - Rotational screw ($\|\omega\| = 1$): $\mathcal{S} = \begin{bmatrix} \omega \\ -\omega \times q \end{bmatrix}$ where q is a point on the axis
  - Translational screw ($\omega = 0$): $\mathcal{S} = \begin{bmatrix} 0 \\ v \end{bmatrix}$ where $\|v\| = 1$
- **SE(2) Screw**: $\mathcal{S} = \begin{bmatrix} \omega_z \\ v_x \\ v_y \end{bmatrix} \in \mathbb{R}^3$
  - Pure rotation ($|\omega_z| = 1$): Center of rotation at $(c_x, c_y) = (-v_y/\omega_z, v_x/\omega_z)$

– Pure translation ($\omega_z = 0$): $\mathcal{S} = \begin{bmatrix} 0 \\ v_x \\ v_y \end{bmatrix}$ where $\sqrt{v_x^2 + v_y^2} = 1$

Exponential Coordinates: Any rigid body displacement can be represented as screw motion:

$$T = e^{[\mathcal{S}]\theta}$$

where $[\mathcal{S}] \in \mathfrak{se}(3)$ or $\mathfrak{se}(2)$ is the matrix representation of the screw, and θ is the magnitude.

Rodrigues' Formula:

- **SE(3)**: For $\mathcal{S} = \begin{bmatrix} \omega \\ v \end{bmatrix}$ with $\|\omega\| = 1$,

$$e^{[\mathcal{S}]\theta} = \begin{bmatrix} e^{[\omega]\theta} & (I\theta + (1 - \cos\theta)[\omega] + (\theta - \sin\theta)[\omega]^2)v \\ 0 & 1 \end{bmatrix}$$

- **SE(2)**: For $\mathcal{S} = \begin{bmatrix} \omega_z \\ v_x \\ v_y \end{bmatrix}$,

$$e^{[\mathcal{S}]\theta} = \begin{bmatrix} \cos(\omega_z\theta) & -\sin(\omega_z\theta) & \frac{v_x \sin(\omega_z\theta) + v_y(1-\cos(\omega_z\theta))}{\omega_z} \\ \sin(\omega_z\theta) & \cos(\omega_z\theta) & \frac{v_y \sin(\omega_z\theta) - v_x(1-\cos(\omega_z\theta))}{\omega_z} \\ 0 & 0 & 1 \end{bmatrix}$$

Velocity from Exponential Coordinates: The body twist is:

$$\mathcal{V} = \mathcal{S}\dot{\theta}$$

This relationship connects the configuration space velocity $\dot{\theta}$ to the geometric velocity (twist) $\mathcal{V}$.

Product of Exponentials (POE): Forward kinematics can be expressed as:

$$T(\theta) = e^{[\mathcal{S}_1]\theta_1} e^{[\mathcal{S}_2]\theta_2} \cdots e^{[\mathcal{S}_n]\theta_n} M$$

where $\mathcal{S}_i$ are the joint screws at zero configuration and M is the home configuration.

A.5 Summary of Key Notation

Symbol	Description
$SE(3)$, $SE(2)$	Special Euclidean groups (spatial transformations)
T_{ab}	Transformation from frame $\{b\}$ to frame $\{a\}$
$\mathcal{V}$, ${}^b\mathcal{V}_a$	Twist (body/spatial velocity)
$\mathcal{F}$	Wrench (generalized force)
$\mathcal{S}$, $\mathcal{B}_i$	Screw axis (spatial/body frame)
$[Ad_T]$	Adjoint matrix for twist transformation
$\{s\}$, $\{l\}$, $\{r\}$	Spatial, left, right reference frames
$\dot{q}_{\text{obj}}$	Object's internal joint velocity

Table 2: Summary of mathematical notation used throughout the paper.

Reference: Notation follows Park & Lynch (2017), *Modern Robotics*.

B Impedance Control on SE(3)

This appendix provides the geometric foundations underlying SWIVL's twist-driven impedance control (Section 3.4). We derive the classical SE(3) impedance formulation from energy principles, identify the nonlinear pose-error Jacobian $J_{\mathcal{E}}$ as a key complexity, and show how SWIVL's approach bypasses this challenge by incorporating pose errors directly into reference twists.

B.1 Notation

We adopt the following notation consistent with the main text:

- $T_{sb} = (R_{sb}, p_{sb}) \in \mathrm{SE}(3)$: Current end-effector pose (body frame $\{b\}$ w.r.t. spatial frame $\{s\}$)
- $T_{sd} = (R_{sd}, p_{sd}) \in \mathrm{SE}(3)$: Desired end-effector pose (desired frame $\{d\}$)
- $T_{bd} = T_{sb}^{-1} T_{sd}$: Relative transformation from current to desired frame
- ${}^b\mathcal{V}_b = (\omega_b, v_b) \in \mathbb{R}^6 \cong \mathfrak{se}(3)$: Current body twist
- ${}^d\mathcal{V}_d = (\omega_d, v_d)$: Desired body twist expressed in frame $\{d\}$
- $\mathcal{E} \in \mathbb{R}^6$: Weighted pose error vector
- $\mathcal{F} \in \mathbb{R}^6 \cong \mathfrak{se}(3)^*$: Wrench (moment, force)
- $\alpha \in \mathbb{R}^+$: Characteristic length weighting rotation vs. translation
- $[\cdot]_\times$: Skew-symmetric matrix representation of $\mathbb{R}^3$

B.2 Metric Tensor on $\mathfrak{se}(3)$

A geometrically consistent impedance controller requires a proper notion of distance on SE(3). We begin by defining an inner product on $\mathfrak{se}(3)$ with clear physical interpretation.

B.2.1 Inner Product as Kinetic Energy

For twists $\mathcal{V}_1 = (\omega_1, v_1)$ and $\mathcal{V}_2 = (\omega_2, v_2)$ in $\mathfrak{se}(3)$, we define the inner product:

$$\langle \mathcal{V}_1, \mathcal{V}_2 \rangle_G = \frac{\alpha^2}{2} \mathrm{tr}([\omega_1]_\times^\top [\omega_2]_\times) + v_1^\top v_2 = \alpha^2 \omega_1^\top \omega_2 + v_1^\top v_2 = \mathcal{V}_1^\top G \mathcal{V}_2 \tag{39}$$

where $G = \mathrm{diag}(\alpha^2 I_3, I_3)$ is the metric tensor. This corresponds to the kinetic energy of an isotropic rigid body with unit mass and moment of inertia $\alpha^2 I_3$:

$$K = \frac{1}{2} \langle \mathcal{V}, \mathcal{V} \rangle_G = \frac{1}{2} \mathcal{V}^\top G \mathcal{V} \tag{40}$$

Physical Interpretation. The characteristic length α acts as the radius of gyration, representing the ratio between rotational and translational inertia. This metric extends to a left-invariant Riemannian metric on SE(3), whose geodesics correspond to free rigid body motion—providing a dynamically consistent notion of distance on the configuration manifold.

B.2.2 Riemannian Metric Extension

We extend the inner product defined on $\mathfrak{se}(3)$ to a left-invariant Riemannian metric on SE(3). For tangent vectors $\dot{T}_1, \dot{T}_2 \in T_T\mathrm{SE}(3)$:

$$\langle \dot{T}_1, \dot{T}_2 \rangle_T = \langle T^{-1}\dot{T}_1, T^{-1}\dot{T}_2 \rangle_G \tag{41}$$

B.2.3 Geodesics and Action Minimization

A geodesic minimizes the action integral along the manifold. For a curve $T(t) \in \mathrm{SE}(3)$ with body twist ${}^b\mathcal{V}_b(t) = (\omega(t), v(t))$, the action integral is:

$$S = \int_{t_0}^{t_f} \langle {}^b\mathcal{V}_b(t), {}^b\mathcal{V}_b(t) \rangle_G \, dt = \int_{t_0}^{t_f} \left(\alpha^2 \|\omega(t)\|^2 + \|v(t)\|^2 \right) dt \tag{42}$$

The Euler-Poincaré equations for the decoupled metric $G = \mathrm{diag}(\alpha^2 I_3, I_3)$ yield:

$$\alpha^2 \dot{\omega} + \omega \times (\alpha^2 \omega) = 0 \quad \Rightarrow \quad \dot{\omega} = 0 \tag{43}$$

$$\dot{v} + \omega \times v = 0 \quad \Rightarrow \quad \dot{v} = -\omega \times v \tag{44}$$

Solving with initial conditions $\omega(0) = \omega_0$ and $v(0) = v_0$:

- **Rotation**: $\omega(t) = \omega_0$ (constant angular velocity about a fixed axis in body frame)
- **Translation**: $v(t) = e^{-[\omega_0]_\times t} v_0 = R(t)^\top v_0$ where $R(t) = e^{[\omega_0]_\times t}$

These solutions describe the motion of an isotropic rigid body: constant angular velocity with linear velocity maintaining constant direction in the spatial frame.

B.2.4 Geodesic Distance and Weighted Pose Error

The geodesic distance between poses T_{sb} and T_{sd} under this metric is:

$$d^2(T_{sb}, T_{sd}) = \alpha^2 \|e_R\|^2 + \|e_p\|^2 \tag{45}$$

where the pose error components in body frame are:

$$e_R = \log(R_{sb}^\top R_{sd})^\vee \in \mathbb{R}^3, \quad e_p = R_{sb}^\top (p_{sd} - p_{sb}) \in \mathbb{R}^3 \tag{46}$$

The **weighted pose error vector** (matching Eq. (6) in the main text):

$$\mathcal{E} = \begin{pmatrix} \alpha\, e_R \\ e_p \end{pmatrix} \in \mathbb{R}^6 \tag{47}$$

satisfies $\|\mathcal{E}\|^2 = d^2(T_{sb}, T_{sd})$, providing a Euclidean representation whose norm equals the SE(3) geodesic distance.

B.3 Virtual Mass-Spring-Damper System on SE(3)

Classical impedance control designs a virtual mechanical system that governs the end-effector's interaction behavior. We derive this system from energy principles.

B.3.1 Potential Energy of Pose Error

We define the potential energy as a quadratic function of the weighted pose error using a symmetric positive semi-definite stiffness matrix $K \in \mathbb{R}^{6\times 6}$:

$$P(\mathcal{E}) = \frac{1}{2} \mathcal{E}^\top K\, \mathcal{E} \tag{48}$$

This potential energy reaches its minimum ($P = 0$) when $\mathcal{E} = 0$, i.e., when the current pose matches the desired pose.

B.3.2 Pose-Error Jacobian and Elastic Wrench Derivation

We derive the elastic wrench using the **power duality principle**. The key is to establish the relationship between the rate of change of the weighted pose error and the body twist through an error Jacobian matrix.

For regulation tasks with static desired pose ($\dot{T}_{sd} = 0$), we derive the error time derivatives in terms of body twist ${}^b\mathcal{V}_b = (\omega_b, v_b)$.

Translation Error Rate. With $e_p = R_{sb}^\top (p_{sd} - p_{sb})$, using $\dot{R}_{sb} = R_{sb}[\omega_b]_\times$ and $\dot{p}_{sb} = R_{sb} v_b$:

$$\begin{aligned} \dot{e}_p &= \frac{d}{dt}(R_{sb}^\top)(p_{sd} - p_{sb}) + R_{sb}^\top(\dot{p}_{sd} - \dot{p}_{sb}) \\ &= (R_{sb}[\omega_b]_\times)^\top (p_{sd} - p_{sb}) - R_{sb}^\top R_{sb} v_b \\ &= [\omega_b]_\times^\top e_p - v_b = -[\omega_b]_\times e_p - v_b = [e_p]_\times \omega_b - v_b \end{aligned} \tag{49}$$

where we used $[\omega_b]_\times^\top = -[\omega_b]_\times$ and the identity $-\omega_b \times e_p = [e_p]_\times \omega_b$.

Rotation Error Rate. Let $R_{\text{err}} = R_{sb}^\top R_{sd}$ so that $e_R = \log(R_{\text{err}})^\vee$. Differentiating for static $\dot{R}_{sd} = 0$:

$$\dot{R}_{\text{err}} = \dot{R}_{sb}^\top R_{sd} = (R_{sb}[\omega_b]_\times)^\top R_{sd} = -[\omega_b]_\times R_{sb}^\top R_{sd} = -[\omega_b]_\times R_{\text{err}} \tag{50}$$

From Lie group theory, if $\dot{R} = [\omega_s]_\times R$ then $\dot{\theta} = J_l^{-1}(\theta)\,\omega_s$, where J_l is the left Jacobian of SO(3):

$$J_l(\theta) = I + \frac{1-\cos\|\theta\|}{\|\theta\|^2}[\theta]_\times + \frac{\|\theta\| - \sin\|\theta\|}{\|\theta\|^3}[\theta]_\times^2 \tag{51}$$

Here $\omega_s = -\omega_b$, so:

$$\dot{e}_R = -J_l^{-1}(e_R)\,\omega_b \tag{52}$$

Weighted Error Jacobian. The weighted error rate is:

$$\dot{\mathcal{E}} = \begin{pmatrix} \alpha\dot{e}_R \\ \dot{e}_p \end{pmatrix} = \begin{pmatrix} -\alpha J_l^{-1}(e_R)\,\omega_b \\ [e_p]_\times\,\omega_b - v_b \end{pmatrix} = -J_{\mathcal{E}}\,{}^b\mathcal{V}_b \tag{53}$$

with the **pose-error Jacobian**:

$$\boxed{J_{\mathcal{E}} = \begin{pmatrix} \alpha J_l^{-1}(e_R) & 0_{3\times3} \\ -[e_p]_\times & I_3 \end{pmatrix} \in \mathbb{R}^{6\times6}} \tag{54}$$

Elastic Wrench via Power Duality. The rate of change of potential energy is:

$$\dot{P} = \frac{\partial P}{\partial \mathcal{E}}^\top \dot{\mathcal{E}} = (K\mathcal{E})^\top \dot{\mathcal{E}} = (K\mathcal{E})^\top(-J_{\mathcal{E}}\,{}^b\mathcal{V}_b) = (-J_{\mathcal{E}}^\top K\mathcal{E})^\top\,{}^b\mathcal{V}_b \tag{55}$$

By power duality, we define the elastic wrench $\mathcal{F}_{\text{elastic}}$ such that $\dot{P} = {}^b\mathcal{V}_b^\top \mathcal{F}_{\text{elastic}}$, yielding:

$$\boxed{\mathcal{F}_{\text{elastic}} = -J_{\mathcal{E}}^\top K\mathcal{E}} \tag{56}$$

Explicit Components. Expanding with $\mathcal{E} = (\alpha e_R, e_p)^\top$ and $K = \left(\begin{smallmatrix} K_{RR} & K_{Rp} \\ K_{pR} & K_{pp} \end{smallmatrix}\right)$:

$$\mathcal{F}_{\text{elastic}} = \begin{pmatrix} m_{\text{elastic}} \\ f_{\text{elastic}} \end{pmatrix}, \quad \begin{aligned} m_{\text{elastic}} &= -\alpha J_l^{-\top}(e_R)(K_{RR}\alpha e_R + K_{Rp}e_p) - e_p \times (K_{pR}\alpha e_R + K_{pp}e_p) \\ f_{\text{elastic}} &= -K_{pR}\alpha e_R - K_{pp}e_p \end{aligned} \tag{57}$$

B.3.3 Twist Error and Kinetic Energy

Given current body twist ${}^b\mathcal{V}_b$ and desired body twist ${}^d\mathcal{V}_d$, we compute their difference in the current body frame using the Adjoint map. For $T_{bd} = T_{sb}^{-1}T_{sd}$ with $R_{bd} = R_{sb}^\top R_{sd}$ and $p_{bd} = R_{sb}^\top(p_{sd} - p_{sb})$:

$$\mathrm{Ad}_{T_{bd}} = \begin{pmatrix} R_{bd} & 0 \\ [p_{bd}]_\times R_{bd} & R_{bd} \end{pmatrix} \tag{58}$$

The twist error in the body frame is:

$$\xi = {}^b\mathcal{V}_d - {}^b\mathcal{V}_b = \mathrm{Ad}_{T_{bd}}\,{}^d\mathcal{V}_d - {}^b\mathcal{V}_b \tag{59}$$

We define the **kinetic energy of the virtual system** as a quadratic function of the twist error:

$$K_{\text{virtual}}(\xi) = \frac{1}{2}\xi^\top M\xi \tag{60}$$

where $M \in \mathbb{R}^{6\times6}$ is the positive-definite virtual mass (inertia) matrix.

Assuming M is constant in the body frame, the rate of change of kinetic energy is:

$$\dot{K}_{\text{virtual}} = \frac{d}{dt}\left(\frac{1}{2}\xi^\top M\xi\right) = \xi^\top M\dot{\xi} \tag{61}$$

The term $M\dot{\xi}$ corresponds to the **inertial wrench**, analogous to ma in Newton's second law.

B.3.4 Complete Virtual Dynamics via Power Balance

The complete virtual system follows from the **power balance equation**. The total energy $E = K_{\text{virtual}} + P$ evolves according to:

$$\dot{E} = P_{\text{ext}} - P_{\text{diss}} \tag{62}$$

where:

- $P_{\text{ext}} = \mathcal{F}_{\text{ext}}^{\top}\xi$: External power input (work done by external wrench on the virtual system)
- $P_{\text{diss}} = \xi^{\top} D\xi$: Dissipated power due to damping ($D \in \mathbb{R}^{6\times 6}$ symmetric positive-definite)

Expanding the power balance:

$$\frac{d}{dt}\left(\frac{1}{2}\xi^{\top} M\xi\right) + \frac{d}{dt}\left(\frac{1}{2}\mathcal{E}^{\top} K\mathcal{E}\right) = \mathcal{F}_{\text{ext}}^{\top}\xi - \xi^{\top} D\xi \tag{63}$$

Using $\dot{K}_{\text{virtual}} = \xi^{\top} M\dot{\xi}$ and $\dot{P} = -\mathcal{F}_{\text{elastic}}^{\top}{}^{b}\mathcal{V}_b = (J_{\mathcal{E}}^{\top} K\mathcal{E})^{\top}{}^{b}\mathcal{V}_b$:

$$\xi^{\top}\left[M\dot{\xi} + J_{\mathcal{E}}^{\top} K\mathcal{E}\right] = \xi^{\top}\left[\mathcal{F}_{\text{ext}} - D\xi\right] \tag{64}$$

Since this must hold for all ξ, we obtain the **virtual mass-spring-damper dynamics**:

$$\boxed{M\dot{\xi} + D\xi + J_{\mathcal{E}}^{\top} K\mathcal{E} = \mathcal{F}_{\text{ext}}} \tag{65}$$

Physical Interpretation:

- $M\dot{\xi}$: Inertial term (virtual mass effect resisting acceleration)
- $D\xi$: Damping term (energy dissipation proportional to velocity error)
- $J_{\mathcal{E}}^{\top} K\mathcal{E}$: Elastic wrench (restoring force toward desired pose)
- $\mathcal{F}_{\text{ext}}$: External excitation from environment contact

B.4 The Pose-Error Jacobian Problem

The Jacobian $J_{\mathcal{E}}$ in Eq. (54) introduces significant complexity:

1. **Nonlinear transcendental functions**: $J_l^{-1}(e_R)$ involves $\sin\|e_R\|$, $\cos\|e_R\|$, and $\|e_R\|^{-1}$ terms that are expensive to compute and differentiate.
2. **Configuration-dependent coupling**: The $-[e_p]_{\times}$ block couples rotation and translation in ways that depend on the current pose error.
3. **Approximation failure**: The common approximation $J_l^{-1}(e_R) \approx I_3$ (valid for small $\|e_R\| \ll 1$) fails in bimanual manipulation where constraint violations can cause large trajectory deviations.
4. **Non-diagonal stiffness requirements**: Object-aware motion decomposition requires non-diagonal K matrices, making the full $J_{\mathcal{E}}^{\top} K\mathcal{E}$ computation unavoidable.

This nonlinearity is the core challenge referenced in Section 3.1 (**C3**) that SWIVL addresses.

B.5 Impedance Controller Implementation

B.5.1 Operational Space Dynamics

The robot's joint space dynamics:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau - J_b^{\top}\mathcal{F}_{\text{ext}} \tag{66}$$

transform to operational space via the body Jacobian $J_b(q)$ with ${}^{b}\mathcal{V}_b = J_b(q)\dot{q}$:

$$\Lambda_b(q)\dot{\mathcal{V}}_b + \mu_b(q,\dot{q}) + \gamma_b(q) = \mathcal{F}_{\text{cmd}} - \mathcal{F}_{\text{ext}} \tag{67}$$

where:

- $\Lambda_b = (J_b M^{-1} J_b^\top)^{-1}$: Operational space inertia (symmetric positive-definite)
- $\mu_b = \Lambda_b J_b M^{-1} C\dot{q} - \Lambda_b \dot{J}_b \dot{q}$: Coriolis/centrifugal wrench
- $\gamma_b = \Lambda_b J_b M^{-1} g$: Gravity wrench in body frame
- $\mathcal{F}_{\text{cmd}}$: Control wrench, related to joint torques by $\tau = J_b^\top \mathcal{F}_{\text{cmd}}$

B.5.2 Controller Design via Virtual-Robot Coupling

To achieve desired impedance behavior, we couple the virtual system dynamics with the robot's operational space dynamics. The key is to match the closed-loop robot behavior to the virtual mass-spring-damper system.

We have two dynamic systems:

- **Virtual System**: $M\dot{\xi} + D\xi + J_{\mathcal{E}}^\top K\mathcal{E} = \mathcal{F}_{\text{ext}}$
- **Robot Dynamics**: $\Lambda_b \dot{\mathcal{V}}_b + \mu_b + \gamma_b = \mathcal{F}_{\text{cmd}} - \mathcal{F}_{\text{ext}}$

Recall $\xi = {}^b\mathcal{V}_d - {}^b\mathcal{V}_b$, so $\dot{\xi} = {}^b\dot{\mathcal{V}}_d - {}^b\dot{\mathcal{V}}_b$.

Step 1: Solve for $\dot{\xi}$ from virtual system:

$$\dot{\xi} = -M^{-1}(D\xi + J_{\mathcal{E}}^\top K\mathcal{E} - \mathcal{F}_{\text{ext}}) \tag{68}$$

Step 2: Substitute into $\dot{\xi} = {}^b\dot{\mathcal{V}}_d - {}^b\dot{\mathcal{V}}_b$ and solve for ${}^b\dot{\mathcal{V}}_b$:

$${}^b\dot{\mathcal{V}}_b = {}^b\dot{\mathcal{V}}_d + M^{-1}(D\xi + J_{\mathcal{E}}^\top K\mathcal{E} - \mathcal{F}_{\text{ext}}) \tag{69}$$

Step 3: Substitute into robot dynamics:

$$\Lambda_b \left[{}^b\dot{\mathcal{V}}_d + M^{-1}(D\xi + J_{\mathcal{E}}^\top K\mathcal{E} - \mathcal{F}_{\text{ext}}) \right] + \mu_b + \gamma_b = \mathcal{F}_{\text{cmd}} - \mathcal{F}_{\text{ext}} \tag{70}$$

Step 4: Solve for control wrench:

$$\boxed{\mathcal{F}_{\text{cmd}} = \Lambda_b M^{-1}(D\xi + J_{\mathcal{E}}^\top K\mathcal{E}) + \Lambda_b {}^b\dot{\mathcal{V}}_d + \mu_b + \gamma_b + (I - \Lambda_b M^{-1})\mathcal{F}_{\text{ext}}} \tag{71}$$

Controller Components:

- $\Lambda_b M^{-1}(D\xi + J_{\mathcal{E}}^\top K\mathcal{E})$: Impedance feedback with weighted error Jacobian
- $\Lambda_b {}^b\dot{\mathcal{V}}_d$: Feedforward acceleration term
- $\mu_b + \gamma_b$: Coriolis and gravity compensation
- $(I - \Lambda_b M^{-1})\mathcal{F}_{\text{ext}}$: External force compensation (inertia-dependent)

B.5.3 Simplified Cases

Case 1: $M = \Lambda_b$ (virtual mass matches robot inertia):

$$\mathcal{F}_{\text{cmd}} = D\xi + J_{\mathcal{E}}^\top K\mathcal{E} + \Lambda_b {}^b\dot{\mathcal{V}}_d + \mu_b + \gamma_b \tag{72}$$

Case 2: $M = \Lambda_b$ **and** ${}^b\dot{\mathcal{V}}_d = 0$ (regulation task):

$$\mathcal{F}_{\text{cmd}} = D\xi + J_{\mathcal{E}}^\top K\mathcal{E} + \mu_b + \gamma_b \tag{73}$$

Case 3: Small rotation errors ($\|e_R\| \ll 1$ so $J_l^{-1} \approx I_3$) with isotropic stiffness ($K = kI_6$):

$$J_{\mathcal{E}} \approx \begin{pmatrix} \alpha I_3 & 0 \\ -[e_p]_\times & I_3 \end{pmatrix}, \quad \mathcal{F}_{\text{cmd}} = D\xi + K_\alpha \mathcal{E} + \mu_b + \gamma_b \tag{74}$$

where $K_\alpha = \text{diag}(\alpha k I_3, k I_3)$ recovers the familiar linear impedance form.

B.6 SWIVL's Twist-Driven Approach

SWIVL bypasses $J_{\mathcal{E}}$ by incorporating pose errors directly into reference twists rather than computing elastic wrenches explicitly.

B.6.1 Reference Twist with Pose Error Correction

Recall from Section 3.3 that SWIVL constructs a reference twist (Eq. (5)):

$$\mathcal{V}_i^{\text{ref}} = \text{Ad}_{T_{b_i d_i}} \mathcal{V}_i^{\text{des}} + k_{p_i} \mathcal{E}_i \tag{75}$$

where $\mathcal{E}_i = (\alpha e_{R_i}, e_{p_i})^\top$ is the weighted pose error.

B.6.2 Controller Formulation

SWIVL's commanded wrench (Eq. (12)) is:

$$\mathcal{F}_{\text{cmd},i} = K_{d_i}(\mathcal{V}_i^{\text{ref}} - \mathcal{V}_i) + \mu_{b,i} + \gamma_{b,i} \tag{76}$$

where $K_{d_i} = G(P_{i,\|} d_{i,\|} + P_{i,\perp} d_{i,\perp})$ is the decomposed damping matrix.

B.6.3 Equivalence to SE(3) Impedance

Expanding Eq. (76) with the reference twist definition:

$$\begin{aligned} \mathcal{F}_{\text{cmd},i} &= K_{d_i}(\text{Ad}_{T_{b_i d_i}} \mathcal{V}_i^{\text{des}} - \mathcal{V}_i + k_{p_i} \mathcal{E}_i) + \mu_{b,i} + \gamma_{b,i} \\ &= K_{d_i} \xi_i + K_{d_i} k_{p_i} \mathcal{E}_i + \mu_{b,i} + \gamma_{b,i} \end{aligned} \tag{77}$$

where $\xi_i = \text{Ad}_{T_{b_i d_i}} \mathcal{V}_i^{\text{des}} - \mathcal{V}_i$ is the twist error.

Comparing with the classical impedance controller (Eq. (73)):

$$\mathcal{F}_{\text{cmd}} = D\xi + J_{\mathcal{E}}^\top K \mathcal{E} + \mu_b + \gamma_b \tag{78}$$

The correspondence is:

$$\underbrace{K_{d_i}}_{\text{learned } D} \xi_i + \underbrace{K_{d_i} k_{p_i} \mathcal{E}_i}_{\approx J_{\mathcal{E}}^\top K \mathcal{E}} + \mu_{b,i} + \gamma_{b,i} \tag{79}$$

Key Insight. SWIVL replaces the explicit nonlinear term $J_{\mathcal{E}}^\top K \mathcal{E}$ with $K_{d_i} k_{p_i} \mathcal{E}_i$, which:

- Avoids computing $J_l^{-1}(e_R)$ and its configuration-dependent coupling
- Maintains impedance behavior through the pose error correction in the reference twist
- Enables the RL policy to learn appropriate k_{p_i} values that adapt to task requirements

Under small rotation errors where $J_l^{-1}(e_R) \approx I_3$, the approximation $J_{\mathcal{E}} \approx \text{diag}(\alpha I_3, I_3)$ shows that $K_{d_i} k_{p_i} \mathcal{E}_i \approx k_{p_i} G \mathcal{E}_i$, recovering a diagonal stiffness structure. For larger errors, the learned k_{p_i} compensates for the approximation.

C SWIVL Instantiation in SE(2)

While the SWIVL framework presented in Section 3 is formulated for general bimanual manipulation of k-DoF articulated objects in SE(3), our experimental evaluation in Section 4 focuses on SE(2) planar tasks with a
textbfsingle internal joint ($k = 1$). This design choice allows systematic study of force coupling and constraint satisfaction while controlling for the additional complexity of full 3D manipulation. Here we detail how the SE(3) formulation naturally reduces to SE(2) in this 1-DoF setting and how each component of SWIVL is instantiated.

C.1 SE(2) Geometric Formulation (1-DoF Object)

C.1.1 Configuration Space

In SE(2), poses are represented as $(x, y, \theta) \in \mathbb{R}^2 \times SO(2)$, where (x, y) is planar position and θ is orientation around the vertical z-axis. The homogeneous transformation matrix:

$$T \in SE(2): \quad T = \begin{bmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{bmatrix}$$

C.1.2 Twist Space

The Lie algebra $\mathfrak{se}(2)$ consists of planar twists:

$$\mathcal{V} = \begin{bmatrix} \omega_z \\ v_x \\ v_y \end{bmatrix} \in \mathbb{R}^3$$

where $\omega_z \in \mathbb{R}$ is angular velocity around z-axis and $(v_x, v_y) \in \mathbb{R}^2$ is linear velocity in the plane.

C.1.3 Screw Axis in SE(2)

For planar articulated objects with a
textbfsingle kinematic joint, the screw axis $\mathcal{S} = \begin{bmatrix} s_\omega \\ s_v \end{bmatrix}$ reduces to:

$$\mathcal{S} = \begin{bmatrix} s_\omega \\ s_{v,x} \\ s_{v,y} \end{bmatrix} \in \mathbb{R}^3$$

Joint Type Examples:

- **Revolute joint** (rotation around z-axis): $\mathcal{S} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ (pure rotation)
- **Prismatic joint** (translation along direction $\hat{d}$): $\mathcal{S} = \begin{bmatrix} 0 \\ d_x \\ d_y \end{bmatrix}$ where (d_x, d_y) defines sliding direction

In the general SE(3) formulation, the object Jacobian $J_s(\mathbf{q}_{obj}) \in \mathbb{R}^{6\times k}$ relates internal joint velocities to relative end-effector motion. In our SE(2), 1-DoF setting, this reduces to a single spatial screw axis $\mathcal{S} \in \mathbb{R}^3$ and the kinematic constraint becomes:

$$^s\mathcal{V}_l - {}^s\mathcal{V}_r = \mathcal{S}\,\dot{q}_{obj}, \quad {}^s\mathcal{V}_i \in \mathbb{R}^3, \quad \mathcal{S} \in \mathbb{R}^3, \quad \dot{q}_{obj} \in \mathbb{R}.$$

For each grasp, the corresponding **body-frame joint screw axes** $\mathcal{B}_l, \mathcal{B}_r \in \mathbb{R}^3$ are obtained by transforming $\mathcal{S}$ into the left and right end-effector frames via the SE(2) adjoint (Appendix A). Because the object has a single joint and grasps remain fixed, these body-frame screw axes are **constant in time and independent of the joint configuration** q_{obj}:

$$\mathcal{B}_i = [Ad_{T_{ib}}]\,{}^b\mathcal{S}, \quad i \in \{l, r\}, \quad \mathcal{B}_i \text{ fixed for a given object.}$$

Thus, the object Jacobians in each body frame collapse to

$$J_l(q_{obj}) = \mathcal{B}_l \in \mathbb{R}^{3\times 1}, \qquad J_r(q_{obj}) = \mathcal{B}_r \in \mathbb{R}^{3\times 1},$$

which no longer depend on q_{obj}.

C.1.4 Wrench Space

Forces and moments in SE(2) are dual to twists:

$$\mathcal{F} = \begin{bmatrix} m_z \\ f_x \\ f_y \end{bmatrix} \in \mathbb{R}^3$$

where m_z is moment around z-axis and (f_x, f_y) are planar forces.

C.1.5 Adjoint Representation

The adjoint transformation for frame changes in SE(2):

$$[Ad_T] = \begin{bmatrix} 1 & 0 & 0 \\ y & \cos\theta & -\sin\theta \\ -x & \sin\theta & \cos\theta \end{bmatrix} \in \mathbb{R}^{3\times 3}$$

Twist transformation between frames:

$${}^s\mathcal{V} = [Ad_{T_{si}}]\mathcal{V}_i, \quad \mathcal{V}_i = [Ad_{T_{si}^{-1}}]{}^s\mathcal{V}$$

C.2 Reference Twist Field Generator in SE(2)

C.2.1 SE(2) Trajectory Smoothing

Given discrete waypoints $\{T_{si}^{des}[\tau]\}_{\tau=0}^{H} = \{(x[\tau], y[\tau], \theta[\tau])\}_{\tau=0}^{H}$ from the high-level planner at 10 Hz, we generate dense trajectories at 100 Hz (Low-Level Policy frequency).

Position Interpolation: Cubic spline interpolation through position waypoints $(x[\tau], y[\tau])$:

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \sum_{j=0}^{3} a_j s^j, \quad s = \frac{t - t_k}{t_{k+1} - t_k}$$

where coefficients $\{a_j\}$ satisfy boundary conditions (positions and velocities at waypoints).

Orientation Interpolation: Circular interpolation on SO(2) ensuring shortest path:

$$\theta(t) = \theta_k + \text{wrap}(\theta_{k+1} - \theta_k) \cdot \phi(s)$$

where $\phi(s) = 3s^2 - 2s^3$ (cubic smoothing), and wrap ensures $|\theta_{k+1} - \theta_k| \leq \pi$.

C.2.2 Body Twist Computation

From smooth trajectory $T_{si}^{des}(t)$, compute desired body twist via time differentiation:

$$\mathcal{V}_i^{des}(t) = \begin{bmatrix} \dot{\theta}(t) \\ \dot{x}(t)\cos\theta(t) + \dot{y}(t)\sin\theta(t) \\ -\dot{x}(t)\sin\theta(t) + \dot{y}(t)\cos\theta(t) \end{bmatrix} \in \mathbb{R}^3$$

C.2.3 Stable Imitation Vector Field

Following Method Eq. (5), the reference twist combines imitation and stability components:

$$\mathcal{V}_i^{\text{ref}}(t, T_{sb_i}) = \text{Ad}_{T_{b_i d_i}} \mathcal{V}_i^{\text{des}}(t) + k_{p_i} \mathcal{E}_i$$

where Ad_T denotes the SE(2) adjoint transformation that maps twists between frames. Since the desired twist $\mathcal{V}_i^{\text{des}}(t)$ is computed in the desired frame $\{d_i\}$, we must transform it to the current body

frame $\{b_i\}$ where the controller operates. The transformation $T_{b_i d_i} = T_{b_i s} T_{s d_i} = (T_{s b_i})^{-1} T_{s d_i}$ represents the relative transformation from the desired frame to the current body frame.

For SE(2), the adjoint transformation is:

$$\mathrm{Ad}_{T_{b_i d_i}} = \begin{bmatrix} 1 & 0 & 0 \\ \Delta y & \cos\Delta\theta & -\sin\Delta\theta \\ -\Delta x & \sin\Delta\theta & \cos\Delta\theta \end{bmatrix} \in \mathbb{R}^{3\times 3}$$

where $(\Delta x, \Delta y, \Delta\theta)$ are the components of $T_{b_i d_i}$.

The pose error term $\mathcal{E}_i \in \mathbb{R}^3$ is given by:

SE(2) Logarithm Map: For pose error $\Delta T = T_{si}^{des} (t^*)^{-1} T_{si}$:

$$[\log(\Delta T)]^\vee = \begin{bmatrix} \Delta\theta \\ \Delta x \cos\theta_{des} + \Delta y \sin\theta_{des} \\ -\Delta x \sin\theta_{des} + \Delta y \cos\theta_{des} \end{bmatrix}$$

where $\Delta x = x - x_{des}$, $\Delta y = y - y_{des}$, $\Delta\theta = \mathrm{wrap}(\theta - \theta_{des})$.

C.3 Bulk-Internal Decomposition via Projection Operators in SE(2)

Following Method Section 3.2.4, SWIVL uses projection operators based on the learned metric tensor $G = \mathrm{diag}(\alpha^2, 1, 1)$ to decompose twists into bulk and internal motion components. This approach enables independent impedance modulation for each component.

C.3.1 Metric Tensor and Inner Product

The SE(2) inner product on $\mathfrak{se}(2)$ is defined using the metric tensor $G \in \mathbb{R}^{3\times 3}$:

$$\langle \mathcal{V}_1, \mathcal{V}_2 \rangle_G = \mathcal{V}_1^\top G \mathcal{V}_2 = \alpha^2 \omega_{1,z} \omega_{2,z} + v_{1,x} v_{2,x} + v_{1,y} v_{2,y}$$

where $\alpha \in \mathbb{R}^+$ is the **learnable characteristic length scale** (part of the RL action space) that weights rotational versus translational components. By learning α, the policy discovers task-appropriate notions of orthogonality for separating bulk versus internal motions.

C.3.2 Projection Operators

For each end-effector $i \in \{l, r\}$ with constant body-frame screw axis $\mathcal{B}_i \in \mathbb{R}^{3\times 1}$ (1-DoF object), we construct orthogonal projection operators:

$$P_{i,\parallel} = \mathcal{B}_i (\mathcal{B}_i^\top G \mathcal{B}_i)^{-1} \mathcal{B}_i^\top G \in \mathbb{R}^{3\times 3} \quad \text{(project onto internal motion)},$$
$$P_{i,\perp} = I_3 - P_{i,\parallel} \in \mathbb{R}^{3\times 3} \quad \text{(project onto bulk motion)}.$$

These operators satisfy:

- $P_{i,\parallel}^\top G = G P_{i,\parallel}$ (G-self-adjoint for internal projection)
- $P_{i,\perp}^\top G = G P_{i,\perp}$ (G-self-adjoint for bulk projection)
- $P_{i,\parallel} + P_{i,\perp} = I_3$ (partition of identity)
- $P_{i,\parallel} P_{i,\perp} = 0$ (orthogonal subspaces under G-metric)

C.3.3 Twist Decomposition

Given reference body twist $\mathcal{V}_i^{\text{ref}} \in \mathbb{R}^3$, decompose into bulk and internal components:

$$\mathcal{V}_{i,\parallel}^{\text{ref}} = P_{i,\parallel} \mathcal{V}_i^{\text{ref}} \in \mathbb{R}^3 \quad \text{(internal motion: range of } \mathcal{B}_i\text{)},$$
$$\mathcal{V}_{i,\perp}^{\text{ref}} = P_{i,\perp} \mathcal{V}_i^{\text{ref}} \in \mathbb{R}^3 \quad \text{(bulk motion: orthogonal complement)}.$$

This decomposition satisfies G-orthogonality: $\langle \mathcal{V}_{i,\parallel}^{\text{ref}}, \mathcal{V}_{i,\perp}^{\text{ref}} \rangle_G = 0$.

Physical Interpretation:

- $\mathcal{V}_{i,\parallel}^{\text{ref}}$: Motion component aligned with the object's kinematic constraint (drives joint articulation)
- $\mathcal{V}_{i,\perp}^{\text{ref}}$: Motion component orthogonal to constraint (drives overall object transport/reorientation)

The Low-Level Policy receives both the full reference twists $\{\mathcal{V}_l^{\text{ref}}, \mathcal{V}_r^{\text{ref}}\}$ and their decomposed components, enabling it to learn task semantics from trajectory structure.

C.3.4 Wrench Decomposition

By duality, wrenches decompose using transposed projection operators. For measured wrench $\mathcal{F}_i \in \mathbb{R}^3$:

$$\mathcal{F}_{i,\parallel} = P_{i,\parallel}^\top \mathcal{F}_i \in \mathbb{R}^3 \quad \text{(productive wrench)},$$
$$\mathcal{F}_{i,\perp} = P_{i,\perp}^\top \mathcal{F}_i \in \mathbb{R}^3 \quad \text{(internal wrench)}.$$

Internal wrench $\mathcal{F}_{i,\perp}$ represents non-productive contact forces that stress the grasp without contributing to joint motion. The RL reward explicitly penalizes $\|\mathcal{F}_{i,\perp}\|_2^2$ to minimize harmful internal forces (Section 3.3.2).

Output Summary: The Reference Twist Field Generator produces at each timestep (100 Hz):

$$\{\mathcal{V}_l^{\text{ref}}, \mathcal{V}_r^{\text{ref}}, \mathcal{B}_l, \mathcal{B}_r\}$$

where individual reference twists $\mathcal{V}_i^{\text{ref}} = \text{Ad}_{T_{b_i d_i}} \mathcal{V}_i^{\text{des}} + k_{p_i} \mathcal{E}_i$ are computed from the stable imitation vector field (Eq. (5) in Method Section 3.2.2). The Low-Level Policy applies projection operators $P_{i,\parallel}$ and $P_{i,\perp}$ (parameterized by learned α) to decompose these into bulk and internal components. Constant screw axes $\mathcal{B}_l, \mathcal{B}_r$ encode the 1-DoF constraint structure.

C.4 Low-Level Policy Architecture for SE(2)

C.4.1 Observation Space

Following Method Section 3.2.3, the SE(2) policy observes:

1. Reference Twists ($\mathbb{R}^6$):

- $\mathcal{V}_l^{\text{ref}}, \mathcal{V}_r^{\text{ref}} \in \mathbb{R}^3$: Reference motions computed by the Reference Twist Field Generator (Layer 2) at the current time t and current end-effector poses T_{sb_l}, T_{sb_r} (6-dim)

2. Object Constraints ($\mathbb{R}^6$):

- $\mathcal{B}_l, \mathcal{B}_r \in \mathbb{R}^3$: Body-frame screw axes defining the object's allowable internal motion directions at each end-effector (6-dim)

3. Wrench Feedback ($\mathbb{R}^6$):

- $\mathcal{F}_l, \mathcal{F}_r \in \mathbb{R}^3$: Body wrenches measured at the end-effectors (6-dim)

4. Proprioception ($\mathbb{R}^{12}$):

- End-effector poses: $T_{sb_l}, T_{sb_r} \in \text{SE}(2)$, represented as $(x_i, y_i, \theta_i) \in \mathbb{R}^3 \times 2$ (6-dim)
- End-effector body twists: $\mathcal{V}_l, \mathcal{V}_r \in \mathbb{R}^3$, represented as $(\omega_{z,i}, v_{x,i}, v_{y,i}) \in \mathbb{R}^3 \times 2$ (6-dim)

Total: $o_t \in \mathbb{R}^{30}$ **(6+6+6+12=30-dim)**

Note: The policy receives the core physical observations that directly parameterize the impedance controller. The policy network internally computes bulk-internal decomposition of reference twists and measured wrenches using projection operators $P_{i,\|}$ and $P_{i,\perp}$ parameterized by the learned metric tensor $G = \mathrm{diag}(\alpha^2, 1, 1)$.

C.4.2 Action Space

Following the SE(3) formulation in Method Section 3.2.3, the SE(2) policy outputs impedance modulation variables adapted for the planar, 1-DoF setting:

Action Space:

$$a_t = (d_{l,\|}, d_{r,\|}, d_{l,\perp}, d_{r,\perp}, k_{p_l}, k_{p_r}, \alpha) \in \mathbb{R}^7$$

where:

- $d_{l,\|}, d_{r,\|} \in \mathbb{R}^+$: Per-arm damping coefficients for internal motion (parallel to screw axis)
- $d_{l,\perp}, d_{r,\perp} \in \mathbb{R}^+$: Per-arm damping coefficients for bulk motion (orthogonal to screw axis)
- $k_{p_l}, k_{p_r} \in \mathbb{R}^+$: Per-arm stiffness gains for the stability term $k_{p_i}\mathcal{E}_i$ in the reference vector field (Eq. (5))
- $\alpha \in \mathbb{R}^+$: Learnable characteristic length scale that defines the metric tensor $G = \mathrm{diag}(\alpha^2, 1, 1)$ for the SE(2) inner product, enabling task-appropriate orthogonal decomposition of twists and wrenches

Note: This maintains the full SE(3) action space structure $(d_{l,\|}, d_{r,\|}, d_{l,\perp}, d_{r,\perp}, k_{p_l}, k_{p_r}, \alpha) \in \mathbb{R}^7$, preserving the ability to independently modulate compliance for each arm. Gripper commands are omitted as grippers remain closed throughout episodes.

C.4.3 SE(2) Screw-decomposed Controller

Following the SE(3) controller formulation in Method Section 3.2.4 (Eq. (8)–(12)), the impedance variables parameterize an SE(2) twist-driven impedance controller:

Orthogonal Projection Operators:

Using the metric tensor $G = \mathrm{diag}(\alpha^2, 1, 1) \in \mathbb{R}^{3\times3}$ and body-frame screw axes $\mathcal{B}_i \in \mathbb{R}^{3\times1}$:

$$P_{i,\|} = \mathcal{B}_i(\mathcal{B}_i^\top G \mathcal{B}_i)^{-1}\mathcal{B}_i^\top G \in \mathbb{R}^{3\times3},$$
$$P_{i,\perp} = I_3 - P_{i,\|} \in \mathbb{R}^{3\times3}$$

where $P_{i,\|}$ projects onto internal motion (range of $\mathcal{B}_i$) and $P_{i,\perp}$ projects onto bulk motion (orthogonal complement).

Damping Matrix Construction:

$$K_{d_i} = G(P_{i,\|}d_{i,\|} + P_{i,\perp}d_{i,\perp}) \in \mathbb{R}^{3\times3}$$

where $d_{i,\|}$ and $d_{i,\perp}$ denote the per-arm damping coefficients ($d_{l,\|}, d_{l,\perp}$ for left arm, $d_{r,\|}, d_{r,\perp}$ for right arm). This allows independent damping modulation for each arm: $d_{i,\|}$ controls compliance along the object's kinematic constraint (internal motion), while $d_{i,\perp}$ controls compliance orthogonal to it (bulk motion).

Commanded Wrench:

$$\mathcal{F}_{\mathrm{cmd},i} = K_{d_i}(\mathcal{V}_i^{\mathrm{ref}} - \mathcal{V}_i) + \mu_{b,i} \in \mathbb{R}^3$$

where $\mathcal{V}_i^{\mathrm{ref}} = \mathrm{Ad}_{T_{b_i d_i}}\mathcal{V}_i^{\mathrm{des}} + k_{p_i}\mathcal{E}_i$ is the reference twist from Layer 2 (Eq. (5)), $\mu_{b,i} = C_{b,i}(q_i, \dot{q}_i)\dot{q}_i$ accounts for Coriolis/centrifugal terms, and gravity $\gamma_{b,i} = 0$ in planar settings.

Execution:

In our BiarT SE(2) simulation environment (Section 4.1) with direct body wrench control, the commanded wrenches $\mathcal{F}_{\mathrm{cmd},i} \in \mathbb{R}^3$ are directly applied as control inputs to the end-effectors. The simulation environment integrates these wrench commands to update end-effector poses, consistent with the impedance-based control framework.

Kinematic Constraint Satisfaction:

By construction, the projection-based structure ensures the holonomic constraint is satisfied. The reference twists $\mathcal{V}_i^{\text{ref}}$ already respect the constraint through the Reference Twist Field Generator, and the damping matrix K_{d_i} preserves the constraint subspace through its construction from $P_{i,\|}$ and $P_{i,\perp}$.

C.5 Controller Implementation

C.5.1 Direct End-Effector Control

In the BiarT environment, we use direct end-effector wrench control without intermediate joint-space representations. The commanded body wrenches $\mathcal{F}_{\mathrm{cmd},i} \in \mathbb{R}^3$ (computed from the impedance controller above) are directly applied as control inputs to the end-effectors. The simulation environment integrates these wrench commands through forward dynamics to update end-effector poses at each control step. This wrench-based control scheme is appropriate for the planar manipulation tasks and allows us to focus on the core challenges of force coupling and constraint satisfaction while maintaining full consistency with the impedance control framework in Method Section 3.2.4.

Control Frequency: 100 Hz (policy and controller run at the same frequency).

C.6 Reward Function in SE(2)

The reward function for SE(2) specializes the general formulation from Method Section 3.3.2, adapted for planar manipulation with the learned metric tensor $G = \mathrm{diag}(\alpha^2, 1, 1)$:

$$r_t = r_{\text{track}} + r_{\text{safety}} + r_{\text{reg}} + r_{\text{term}}$$

C.6.1 Motion Tracking Reward

Following Method Eq. (29), the tracking reward uses the G-metric to measure velocity error:

$$r_{\text{track}} = -w_{\text{track}} \sum_{i \in \{l,r\}} \|\mathcal{V}_i - \mathcal{V}_i^{\text{ref}}\|_G^2 = -w_{\text{track}} \sum_{i \in \{l,r\}} (\mathcal{V}_i - \mathcal{V}_i^{\text{ref}})^T G (\mathcal{V}_i - \mathcal{V}_i^{\text{ref}})$$

Expanding with $G = \mathrm{diag}(\alpha^2, 1, 1)$ and $\mathcal{V}_i = [\omega_{z,i}, v_{x,i}, v_{y,i}]^T \in \mathbb{R}^3$:

$$\|\mathcal{V}_i - \mathcal{V}_i^{\text{ref}}\|_G^2 = \alpha^2(\omega_{z,i} - \omega_{z,i}^{\text{ref}})^2 + (v_{x,i} - v_{x,i}^{\text{ref}})^2 + (v_{y,i} - v_{y,i}^{\text{ref}})^2$$

This ensures tracking error is measured consistently with the impedance control framework, with adaptive weighting between rotational and translational components via the learned parameter α.

C.6.2 Exponential Safety Reward

Following Method Eq. (30), the safety reward uses an **exponential form** that provides a positive 'alive bonus' when fighting forces are low:

$$r_{\text{safety}} = w_{\text{safety}} \exp\left(-\kappa \sum_{i \in \{l,r\}} \|\mathcal{F}_{i,\perp}\|_{G^{-1}}^2 \right)$$

where $\kappa > 0$ is a decay rate and the wrench norm uses the **dual metric** $G^{-1} = \mathrm{diag}(1/\alpha^2, 1, 1)$:

$$\|\mathcal{F}_{i,\perp}\|_{G^{-1}}^2 = \frac{m_{z,\perp}^2}{\alpha^2} + f_{x,\perp}^2 + f_{y,\perp}^2$$

Mathematical Justification for the Dual Metric. Twists and wrenches are elements of dual vector spaces related through the reciprocal product (virtual power): $P = \mathcal{F}^\top \mathcal{V}$. When the twist space is equipped with metric G, the wrench space naturally inherits the dual metric G^{-1}:

- **Dimensional consistency:** The moment m_z has units [force×length], while forces f_x, f_y have units [force]. Using the Euclidean norm $\|F\|_2^2 = m_z^2 + f_x^2 + f_y^2$ is *dimensionally inconsistent*. The dual metric G^{-1} with m_z^2/α^2 normalizes the moment by the characteristic length, yielding consistent [force^2] units for all terms.
- **Musical isomorphism:** The metric G on $\mathfrak{se}(2)$ induces an isomorphism to the dual space $\mathfrak{se}(2)^*$ via $\mathcal{V} \mapsto G\mathcal{V}$. The corresponding inner product on $\mathfrak{se}(2)^*$ is given by G^{-1}.
- **Adaptive weighting:** Since α is learned, the policy discovers task-appropriate weighting between moment and force penalties.

Design Rationale. A naive quadratic penalty $r_{\text{safety}} = -w \sum_i \|\mathcal{F}_{i,\perp}\|^2$ produces all-negative rewards, which can lead to pathological learning: the agent may learn to intentionally trigger early termination to avoid accumulating negative rewards. The exponential formulation avoids this by providing:

- **Positive baseline:** When $\|\mathcal{F}_{i,\perp}\| \approx 0$, the agent receives $r_{\text{safety}} \approx w_{\text{safety}} > 0$
- **Smooth decay:** As fighting forces increase, the reward smoothly decays toward zero
- **Alive bonus effect:** The positive reward incentivizes the agent to maintain safe operation and survive longer episodes

Wrench Decomposition via Projection Operators. Consistent with Method Section 3.3.2 and the twist decomposition in Layer 4, wrenches decompose using the transpose of twist projection operators. For measured wrench $\mathcal{F}_i = [m_{z,i}, f_{x,i}, f_{y,i}]^T \in \mathbb{R}^3$:

$$\mathcal{F}_{i,\|} = P_{i,\|}^T \mathcal{F}_i, \quad \mathcal{F}_{i,\perp} = P_{i,\perp}^T \mathcal{F}_i = (I_3 - P_{i,\|})^T \mathcal{F}_i$$

where $P_{i,\|} = \mathcal{B}_i(\mathcal{B}_i^T G \mathcal{B}_i)^{-1} \mathcal{B}_i^T G$ and $P_{i,\perp} = I_3 - P_{i,\|}$ are the SE(2) projection operators defined in Section C.3.2.

This decomposition exploits the duality between twist and wrench spaces under the reciprocal product (virtual power). For any $\mathcal{V} \in \text{range}(P_{i,\perp})$, we have $\mathcal{V} = P_{i,\perp}\mathcal{V}'$, and:

$$\mathcal{F}_{i,\|}^T \mathcal{V} = (P_{i,\|}^T \mathcal{F}_i)^T (P_{i,\perp}\mathcal{V}') = \mathcal{F}_i^T P_{i,\|} P_{i,\perp} \mathcal{V}' = 0$$

where the last equality follows from $P_{i,\|} P_{i,\perp} = 0$ (orthogonal projections). Similarly, $\mathcal{F}_{i,\perp}^T \mathcal{V} = 0$ for all $\mathcal{V} \in \text{range}(P_{i,\|})$.

Physical Interpretation:

- $\mathcal{F}_{i,\|}$: Productive wrench that performs work along the object's internal degree of freedom (joint articulation)
- $\mathcal{F}_{i,\perp}$: Internal wrench orthogonal to the kinematic constraint that:
  - Does not contribute to desired object motion (zero virtual power along range$(P_{i,\|})$)
  - Arises from coordination errors between the two arms
  - Represents constraint forces (bearing loads, friction, contact stresses) unrelated to joint actuation
  - Increases unnecessary contact stress and grasp instability
  - Wastes energy and risks hardware damage

C.6.3 Regularization Reward

Following Method Eq. (31), the regularization reward encourages smooth motion:

$$r_{\text{reg}} = -w_{\text{reg}} \sum_{i \in \{l,r\}} \|\dot{\mathcal{V}}_i\|_2^2$$

where $\dot{\mathcal{V}}_i = [\ddot{\theta}_i, \dot{v}_{x,i}, \dot{v}_{y,i}]^T \in \mathbb{R}^3$ is the SE(2) twist acceleration. This reduces energy consumption, joint jerkiness, and Cartesian jerkiness, promoting natural and efficient movements.

C.6.4 Termination Penalty

Following Method Eq. (32), a penalty is applied when episodes terminate due to failure conditions:

$$r_{\text{term}} = \begin{cases} -w_{\text{term}} & \text{if failure termination (grasp drift or wrench limit)} \\ 0 & \text{otherwise} \end{cases}$$

This explicit penalty discourages the agent from learning behaviors that intentionally trigger failure conditions to end episodes early.

C.6.5 Termination Conditions

Episodes terminate immediately (with termination penalty r_{term} applied) under two failure conditions:

(1) Grasp Drift. Grasp stability is monitored via geodesic distance on SE(2):

$$\text{Terminate if: } \exists i \in \{l,r\} \text{ such that } \left\| \left[\log \left((T_{\text{grip},i}^{\text{init}})^{-1} T_{\text{grip},i} \right) \right]^\vee \right\|_G > d_{\max}$$

where $T_{\text{grip},i}^{\text{init}}$ is the initial grasp pose, $T_{\text{grip},i}$ is the current grasp pose, and $d_{\max}$ is the maximum allowable drift threshold. For SE(2), the logarithm map computes planar geodesic distance:

$$[\log(\Delta T)]^\vee = \begin{bmatrix} \Delta\theta \\ \Delta x \cos\theta_{\text{init}} + \Delta y \sin\theta_{\text{init}} \\ -\Delta x \sin\theta_{\text{init}} + \Delta y \cos\theta_{\text{init}} \end{bmatrix}$$

(2) Wrench Limit. External wrenches are monitored to prevent hardware damage. We use the **dual metric** $G^{-1} = \text{diag}(1/\alpha^2, 1, 1)$ to compute wrench magnitude, ensuring consistent weighting between moment and force components:

$$\text{Terminate if: } \exists i \in \{l,r\} \text{ such that } \|\mathcal{F}_i\|_{G^{-1}} > \mathcal{F}_{\max}$$

where the G^{-1}-weighted wrench norm is:

$$\|\mathcal{F}_i\|_{G^{-1}}^2 = \frac{m_{z,i}^2}{\alpha^2} + f_{x,i}^2 + f_{y,i}^2$$

This dual metric is consistent with the twist metric $G = \text{diag}(\alpha^2, 1, 1)$, preserving the reciprocal product (virtual power) relationship: $\langle \mathcal{F}, \mathcal{V} \rangle = \mathcal{F}^\top \mathcal{V}$. The learned characteristic length α thus provides adaptive weighting between moment and force limits.

Excessive wrenches can cause:

- Grasp slippage and object dropping
- Object deformation or damage
- End-effector or joint damage
- Dangerous interactions in real-world deployment

Both conditions represent task failure and trigger the termination penalty, encouraging the agent to learn safe manipulation strategies.

C.7 SE(2) $\rightarrow$ SE(3) Extension Path

The SE(2) experimental validation serves as a controlled study of SWIVL's core principles. Extension to SE(3) is straightforward:

Mathematical Framework:

- All SE(3) formulations in Section 3 directly apply
- Twist space: $\mathfrak{se}(2) \subset \mathfrak{se}(3)$ (3-dim $\rightarrow$ 6-dim)
- Metric tensor: $G = \text{diag}(\alpha^2, 1, 1) \in \mathbb{R}^{3\times3} \rightarrow G = \text{diag}(\alpha^2 I_3, I_3) \in \mathbb{R}^{6\times6}$ (scalar rotation $\rightarrow$ 3D rotation weighting)
- Action space: $(d_{l,\parallel}, d_{r,\parallel}, d_{l,\perp}, d_{r,\perp}, k_{p_l}, k_{p_r}, \alpha) \in \mathbb{R}^7$ (same structure for both SE(2) and SE(3))
- Object Jacobian: $\mathcal{B}_i \in \mathbb{R}^{3\times1} \rightarrow J_i \in \mathbb{R}^{6\times k}$ (single screw axis $\rightarrow$ multi-DoF Jacobian)
- Network architecture scales with input/output dimensions

Engineering Requirements:

- 6-axis F/T sensors (already available on Franka FR3)
- 7-DoF differential IK controller (standard in Franka SDK) for joint-space control
- SE(3) trajectory smoothing (geodesic interpolation, Appendix C)
- Robot proprioception (end-effector poses and twists, object tracking, gripper feedback)

Validation Strategy:

1. SE(2) experiments in BiarT (current work): Isolate force coupling and constraint satisfaction with impedance-based control
2. SE(3) simulation: Validate 6-DoF extension with gravity, collisions, and per-arm impedance modulation
3. Real-world deployment: Franka FR3 dual-arm setup

The BiarT SE(2) results provide strong evidence that SWIVL's principles—learned impedance variables, projection-based motion decomposition, FiLM-based object conditioning, and screw-decomposed control—will transfer to full SE(3) manipulation.

D Learning Settings for SE(2) Implementation

This appendix provides comprehensive implementation details for the SWIVL Low-Level Policy in the SE(2) planar manipulation setting, including network architecture, training configuration, and simulation environment specifications. All experiments are conducted in the BiarT (Bimanual Articulated manipulation) environment described in Section 4.1.

D.1 Network Architecture

The Low-Level Policy $\pi_\theta : \mathcal{O} \rightarrow \Delta(\mathcal{A})$ is implemented as a neural network with object-conditioned multi-stream architecture, employing Feature-wise Linear Modulation (FiLM) to inject object geometric structure throughout all feature processing stages.

D.1.1 Input and Output Specifications

Observation Space: $o_t \in \mathbb{R}^{30}$ (SE(2) planar setting)

- **Reference Twists** (6-dim): $\mathcal{V}_l^{\text{ref}}, \mathcal{V}_r^{\text{ref}} \in \mathbb{R}^3$
- **Object Constraints** (6-dim): Body-frame screw axes $\mathcal{B}_l, \mathcal{B}_r \in \mathbb{R}^3$
- **Wrench Feedback** (6-dim): Body wrenches $\mathcal{F}_l, \mathcal{F}_r \in \mathbb{R}^3$

- **Proprioception** (12-dim): End-effector poses $(x_i, y_i, \theta_i) \in \mathbb{R}^3 \times 2$ (6-dim), body twists $\mathcal{V}_i = (\omega_{z,i}, v_{x,i}, v_{y,i}) \in \mathbb{R}^3 \times 2$ (6-dim)

Note: This corresponds to the SE(2) observation space detailed in Method Section 3.2.3 and Appendix C.

Input Normalization: Each modality is normalized before being fed to its respective encoder to ensure balanced gradients and stable learning:

- **Reference Twists**: Twist components clipped to $[-v_{\max}, v_{\max}]$ then scaled by s_{ref}
- **Object Constraints**: Screw axes are already unit-normalized
- **Wrench Feedback**: Running normalization with exponential moving average: $\hat{\mathcal{F}} = (\mathcal{F} - \mu_{\mathcal{F}})/(\sigma_{\mathcal{F}} + \epsilon)$ where $\mu_{\mathcal{F}}, \sigma_{\mathcal{F}}$ are updated online with decay α_{wrench}
- **Proprioception**: Poses clipped to workspace bounds $[-p_{\max}, p_{\max}] \times [-p_{\max}, p_{\max}] \times [-\pi, \pi]$ then scaled by s_{pose}; body twists clipped to $[-\dot{p}_{\max}, \dot{p}_{\max}]$ then scaled by s_{vel}

Action Space: $a_t \in \mathbb{R}^7$ (SE(2) planar setting)

- Per-arm damping coefficients for internal motion: $d_{l,\parallel}, d_{r,\parallel} \in \mathbb{R}$
- Per-arm damping coefficients for bulk motion: $d_{l,\perp}, d_{r,\perp} \in \mathbb{R}$
- Stiffness gains: $k_{p_l}, k_{p_r} \in \mathbb{R}$
- Characteristic length scale: $\alpha \in \mathbb{R}$

These impedance variables parameterize the SE(2) screw-decomposed controller as detailed in Appendix C.

D.1.2 Multi-Stream Encoder Architecture

Object Structure Encoder (Conditioning Generator):

The object encoder processes kinematic constraint information and generates a shared embedding that is then projected to stream-specific FiLM parameters:

$$h_{obj}^{(1)} = \text{SiLU}(\text{LayerNorm}(W_{obj}^{(1)} x_{obj} + b_{obj}^{(1)})) \in \mathbb{R}^{64},$$
$$e_{obj} = \text{SiLU}(\text{LayerNorm}(W_{obj}^{(2)} h_{obj}^{(1)} + b_{obj}^{(2)})) \in \mathbb{R}^{128}$$

where $x_{obj} \in \mathbb{R}^6$ contains body-frame screw axes $\mathcal{B}_l, \mathcal{B}_r \in \mathbb{R}^3$. The shared object embedding e_{obj} is projected to layer-specific FiLM parameters via lightweight affine transformations:

$$[\gamma_s^{(l)}, \beta_s^{(l)}] = W_{FiLM,s}^{(l)} e_{obj} + b_{FiLM,s}^{(l)} \in \mathbb{R}^{d_s} \times \mathbb{R}^{d_s}$$

where $s \in \{\text{ref}, \text{wrench}, \text{proprio}, \text{fuse}\}$ denotes the stream, l is the layer index, and d_s is the feature dimension of that layer. This ensures dimensional compatibility between FiLM parameters and target features.

Reference Motion Encoder:

Processes reference twists with object-aware feature transformation:

$$h_{ref}^{(0)} = \text{SiLU}(\text{LayerNorm}(W_{ref}^{(1)} x_{ref} + b_{ref}^{(1)})) \in \mathbb{R}^{128},$$
$$h_{ref} = \text{FiLM}(\text{LayerNorm}(W_{ref}^{(2)} h_{ref}^{(0)} + b_{ref}^{(2)}); \gamma_{ref}^{(1)}, \beta_{ref}^{(1)}) \in \mathbb{R}^{128}$$

where $x_{ref} \in \mathbb{R}^6$ contains reference twists $\mathcal{V}_l^{\text{ref}}, \mathcal{V}_r^{\text{ref}}$. The policy network internally computes bulk-internal decomposition using projection operators.

Wrench Encoder:

Processes force-torque sensor feedback with object-aware feature transformation:

$$h_{wrench}^{(0)} = \text{SiLU}(\text{LayerNorm}(W_{wrench}^{(1)} x_{wrench} + b_{wrench}^{(1)})) \in \mathbb{R}^{128},$$
$$h_{wrench} = \text{FiLM}(\text{LayerNorm}(W_{wrench}^{(2)} h_{wrench}^{(0)} + b_{wrench}^{(2)}); \gamma_{wrench}^{(1)}, \beta_{wrench}^{(1)}) \in \mathbb{R}^{128}$$

where $x_{wrench} \in \mathbb{R}^6$ contains body wrenches $\mathcal{F}_l, \mathcal{F}_r$. The policy network internally computes productive-internal wrench decomposition using projection operators.

Proprioception Encoder:

Processes robot state information with higher capacity for rich state representation:

$$h_{proprio}^{(0)} = \text{SiLU}(\text{LayerNorm}(W_{proprio}^{(1)} x_{proprio} + b_{proprio}^{(1)})) \in \mathbb{R}^{128},$$
$$h_{proprio} = \text{FiLM}(\text{LayerNorm}(W_{proprio}^{(2)} h_{proprio}^{(0)} + b_{proprio}^{(2)}); \gamma_{proprio}^{(1)}, \beta_{proprio}^{(1)}) \in \mathbb{R}^{128}$$

where $x_{proprio} \in \mathbb{R}^{12}$ contains end-effector poses and velocities.

D.1.3 Multi-Modal Fusion and Policy Head

Feature Fusion:

Encoded features from all streams are concatenated and fused through object-conditioned layers:

$$\tilde{h} = [h_{ref}, h_{wrench}, h_{proprio}] \in \mathbb{R}^{384},$$
$$h_{fused}^{(1)} = \text{SiLU}(\text{FiLM}(W_{fuse}^{(1)} \tilde{h} + b_{fuse}^{(1)}; \gamma_{fuse}^{(1)}, \beta_{fuse}^{(1)})) \in \mathbb{R}^{256},$$
$$h_{context} = \text{FiLM}(W_{fuse}^{(2)} h_{fused}^{(1)} + b_{fuse}^{(2)}; \gamma_{fuse}^{(2)}, \beta_{fuse}^{(2)}) \in \mathbb{R}^{256}$$

Action Decoder:

The fused context is decoded into action distribution parameters:

$$h_{action} = \text{SiLU}(W_{action}^{(1)} h_{context} + b_{action}^{(1)}) \in \mathbb{R}^{128},$$
$$[\mu, \log \sigma] = W_{action}^{(2)} h_{action} + b_{action}^{(2)} \in \mathbb{R}^{14}$$

where $\mu \in \mathbb{R}^7$ and $\log \sigma \in \mathbb{R}^7$ parameterize a diagonal Gaussian action distribution $\pi_\theta(a|o) = \mathcal{N}(a; \mu(o), \text{diag}(\exp(\log \sigma(o))))$ for the 7-dimensional SE(2) impedance action space $(d_{l,\parallel}, d_{r,\parallel}, d_{l,\perp}, d_{r,\perp}, k_{p_l}, k_{p_r}, \alpha)$. The log standard deviation is clipped to $[\log(0.01), \log(10)]$ to prevent numerical instability.

Positivity Constraint: Since all impedance parameters must be strictly positive ($d_{i,\parallel}, d_{i,\perp}, k_{p_i}, \alpha \in \mathbb{R}^+$) for physical stability, the sampled actions from the Gaussian distribution are passed through a Softplus activation function:

$$a_{\text{final}} = \text{Softplus}(a_{\text{sampled}}) = \log(1 + \exp(a_{\text{sampled}}))$$

This ensures $a_{\text{final}} > 0$ for all components while maintaining differentiability for policy gradient updates. The Softplus function provides smooth gradients near zero, avoiding the non-differentiability issues of ReLU or absolute value, and naturally prevents negative damping or stiffness coefficients that would destabilize the impedance controller.

D.1.4 Architectural Components

FiLM Layer: Feature-wise Linear Modulation applies affine transformation based on object conditioning:

$$\text{FiLM}(h; \gamma^{(obj)}, \beta^{(obj)}) = \gamma^{(obj)} \odot h + \beta^{(obj)}$$

where $\gamma^{(obj)}, \beta^{(obj)} \in \mathbb{R}^d$ are stream- and layer-specific parameters projected from the shared object embedding e_{obj} and modulate features element-wise. This enables object-specific feature transformation throughout the network while maintaining dimensional compatibility.

Activation: SiLU (Swish) for smooth gradients: $\text{SiLU}(x) = x \cdot \sigma(x)$

Normalization: LayerNorm with $\epsilon = 10^{-5}$: $\text{LayerNorm}(x) = \frac{x-\mu}{\sqrt{\sigma^2+\epsilon}} \odot \gamma_{LN} + \beta_{LN}$

Note: The learnable parameters γ_{LN} and β_{LN} in LayerNorm are distinct from the FiLM parameters $\gamma^{(obj)}$ and $\beta^{(obj)}$.

D.2 Training Configuration

D.2.1 Reinforcement Learning Algorithm

We train the Low-Level Policy using Proximal Policy Optimization (PPO) with clipped objective:

$$L^{CLIP}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right]$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|o_t)}{\pi_{\theta_{old}}(a_t|o_t)}$ is the probability ratio and advantages are computed via Generalized Advantage Estimation (GAE).

D.2.2 Hyperparameters

Optimization:

- **Optimizer:** Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$
- **Learning rate:** 3×10^{-4} with linear decay over training
- **Gradient clipping:** Maximum norm 0.5
- **Weight decay:** 10^{-4}

PPO Configuration:

- **Rollout horizon:** 256 steps per worker
- **Batch size:** 4096 transitions per iteration
- **Mini-batch size:** 256 transitions per update
- **Update epochs:** 10 epochs per batch
- **Clip range:** $\epsilon = 0.2$
- **Value loss coefficient:** 0.5
- **Entropy coefficient:** $0.01 \rightarrow 0.001$ (linear annealing)

GAE Configuration:

- **Discount factor:** $\gamma = 0.99$
- **GAE lambda:** $\lambda = 0.95$

Policy Distribution:

- **Type:** Diagonal Gaussian with state-dependent standard deviation
- **Initial log std:** $\log \sigma_0 = -0.5$
- **Action bounds:** $[-10, 10]$ for raw Gaussian samples before Softplus transformation (ensuring final positive actions in practical range $[\text{Softplus}(-10), \text{Softplus}(10)] \approx [4.5 \times 10^{-5}, 10.00]$)

D.2.3 Initialization Strategy

Linear Layers: Xavier initialization with fan-averaging:

$$W \sim \mathcal{U}\left(-\sqrt{\frac{6}{n_{in}+n_{out}}}, \sqrt{\frac{6}{n_{in}+n_{out}}}\right)$$

FiLM Generators: Initialize scale parameters near identity and shift parameters near zero to ensure stable initial conditioning:

$$W_\gamma \sim \mathcal{N}(0, 0.01^2), \quad b_\gamma = 1, \quad W_\beta \sim \mathcal{N}(0, 0.01^2), \quad b_\beta = 0$$

This ensures that FiLM conditioning initially approximates identity transformation, preventing disruption of gradient flow during early training.

Action Head: Small-scale initialization to encourage near-zero initial actions:

$$W^{(2)}_{action} \sim \mathcal{N}(0, 0.01^2), \quad b^{(2)}_{action} = 0$$