SWIVL: 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 SE(2) planar benchmarks involving rigid transport, revolute rotation, and prismatic insertion, demonstrating improved success rates, 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 three joint types: fixed, 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 for denoising-based action generation and ACT 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, RT-2, π_0, and Octo leverage large-scale datasets like Open-X Embodiment 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 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 and impedance control provide foundations for compliant manipulation, regulating motion-force relationships through virtual mass-spring-damper dynamics. Hybrid force/position control 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. 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. 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. While recent work in humanoid control employs MPC and RL-trained low-level policies 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** (Screw-Wrench informed Impedance Variable Learning), 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, $\mathbf{q}_{\mathrm{obj}}, \dot{\mathbf{q}}_{\mathrm{obj}} \in \mathbb{R}^k$ are the joint positions and joint velocities, and the kinematic structure is characterized by the spatial Jacobian $\mathbf{J}_s(\mathbf{q}_{\mathrm{obj}}) \in \mathbb{R}^{6 \times k}$, 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:

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

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.
- $\dot{\mathbf{q}}_{\mathrm{obj}} \in \mathbb{R}^k$ 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 **??**). 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 **??**) 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 (~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 (~10Hz), while the low-level controller requires smooth, dense reference trajectories at high frequency (~100Hz) with pose-error based corrective fields.

A key insight is the **pose space vs. twist space mismatch**: high-level planners naturally output poses $T \in \text{SE}(3)$, 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.

- 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 (Appendix B). 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:

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

where $\text{Ad}_{T_{b_i d_i}}$ 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 **??**), 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 $G = \text{diag}(\alpha^2 I_3, I_3)$ 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 $\alpha^2 I_3$:

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

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.

6

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

Using the metric tensor $G = \mathrm{diag}(\alpha^2 I_3, I_3)$ 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:

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

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

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:

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

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, $\mu_{b,i}$ compensates for Coriolis/centrifugal terms, and $\gamma_{b,i}$ provides gravity compensation (omitted in planar SE(2) settings).

7

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

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

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:

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

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:

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

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

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

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

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

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:

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

This decomposition provides two critical properties:

1. **Independent compliance modulation**: The policy can independently adjust $d_{i,\parallel}$ 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:

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

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

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

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

This orthogonality follows from $P_{i,\parallel}^\top G P_{i,\perp} = 0$, 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**: $\{\mathcal{V}_l^{\mathrm{ref}}, \mathcal{V}_r^{\mathrm{ref}}\} \in \mathfrak{se}(3) \times \mathfrak{se}(3)$
   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:

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

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

This is naturally satisfied by:

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

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

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

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

The parallel component $\mathcal{F}_{i,\parallel}$ 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($P_{i,\parallel}$))
- 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 three objectives for stable, force-compliant manipulation:

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

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}). Penalizes bulk forces:

$$r_{\text{safety}} = -w_{\text{int}} \sum_{i \in \{l,r\}} \|\mathcal{F}_{i,\perp}\|_2^2. \tag{30}$$

By penalizing $\|\mathcal{F}_{i,\perp}\|_2^2$, the policy learns to suppress excessive bulk forces while maintaining necessary internal forces. This wrench decomposition is fully consistent with the twist decomposition framework.

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 Conditions. Episodes terminate if grasp drift exceeds a threshold:

$$\text{Terminate if } \exists i \in \{l,r\} : \left\| \left[\log \left((T_{\text{grip},i}^{\text{init}})^{-1} T_{\text{grip},i} \right) \right]^\vee \right\|_2 > d_{\max}, \tag{32}$$

where $T_{\text{grip},i}^{\text{init}}$ and $T_{\text{grip},i}$ are initial and current grasp poses. This geodesic distance on SE(3) captures both translational and rotational drift, encouraging stable grasps without explicit reward shaping.

Algorithm 1 SWIVL Training

Require: High-Level Policy π_{HL}, object set $\mathcal{O}_{\text{obj}}$
 1: Initialize policy parameters θ, value function parameters ϕ
 2: **for** episode = 1 to N_{episodes} **do**
 3:     Sample object $o \sim \mathcal{O}_{\text{obj}}$; initialize environment
 4:     **for** $t = 1$ to H **do**
 5:         *// Layer 1: High-Level Policy*
 6:         **if** $t \mod f_{HL}^{-1} = 0$ **then**
 7:             Generate action chunk $\{T_{sd_i}[\tau]\}_{\tau=0}^{H_{\text{chunk}}} \leftarrow \pi_{HL}$
 8:         **end if**
 9:         *// Layer 2: Reference Twist Field Generator*
10:         Interpolate action chunk $\rightarrow$ dense trajectory $T_{sd_i}(t)$
11:         Compute desired body twists $\mathcal{V}_i^{\text{des}}(t)$ via Eq. (4)
12:         Apply stable vector field $\rightarrow$ reference twists $\mathcal{V}_i^{\text{ref}}$ via Eq. (5)
13:         Decompose $\mathcal{V}_i^{\text{ref}} \rightarrow$ internal $\mathcal{V}_{i,\parallel}^{\text{ref}}$ and bulk $\mathcal{V}_{i,\perp}^{\text{ref}}$
14:         *// Layer 3: Impedance Modulation Policy*
15:         Observe $o_t = (\mathcal{V}_i^{\text{ref}}, \mathcal{B}_i, \mathcal{F}_i, T_{sb_i}, \mathcal{V}_i)$
16:         Sample $(d_{i,\parallel}, d_{i,\perp}, k_{p_i}, \alpha) \sim \pi_\theta(\cdot|o_t)$
17:         *// Layer 4: Screw-Decomposed Controller*
18:         Compute commanded wrench $\mathcal{F}_{\text{cmd},i}$ via Eq. (12)
19:         Execute joint torque $\tau_{\text{cmd},i} = J_i(\theta_i)^\top \mathcal{F}_{\text{cmd},i}$
20:         *// Collect Experience*
21:         Observe o_{t+1}, compute reward r_t via Eq. (28)
22:         Store transition (o_t, a_t, r_t, o_{t+1})
23:     **end for**
24:     *// PPO Update*
25:     Compute advantages via GAE (Eq. (34))
26:     **for** epoch = 1 to K **do**
27:         Sample mini-batches; update θ via Eq. (33); update ϕ via MSE loss
28:     **end for**
29: **end for**

3.5.4 Policy Architecture and Training

The policy $\pi_\theta : \mathcal{O} \rightarrow \mathcal{A}$ uses a multi-stream architecture with **Feature-wise Linear Modulation (FiLM)** to inject object geometry into feature processing. Different joint types (revolute, prismatic, fixed) require fundamentally different impedance strategies. 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)**:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right], \tag{33}$$

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):

$$\hat{A}_t = \sum_{l=0}^{\infty}(\gamma\lambda)^l \delta_{t+l}, \quad \delta_t = r_t + \gamma V_\phi(o_{t+1}) - V_\phi(o_t), \tag{34}$$

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

4 Experiments

We evaluate SWIVL on bimanual manipulation of a **single-joint articulated object** in an SE(2) planar benchmark environment. While the Method section develops a general SE(3) formulation for k-DoF articulated objects, our experiments specialize this framework to the planar, 1-DoF setting

where the object Jacobians collapse to fixed body-frame screw axes for each gripper. Our experiments address three key questions: (Q1) Does SWIVL improve task success and reduce internal forces compared to imitation learning baselines? (Q2) How do design choices in the reference motion decomposition and impedance modulation affect performance? (Q3) Can SWIVL generalize across diverse high-level planners and novel object instances within this SE(2) benchmark?

Why SE(2) and 1-DoF? Our SE(2) planar environment with single-joint articulated objects is a deliberate design choice that enables controlled, rigorous evaluation of SWIVL's core methodology while maintaining full mathematical consistency with the general framework. This simplification strategy follows established scientific practice of validating fundamental principles in tractable settings before scaling to full complexity.

Rationale for SE(2) planar restriction: By limiting to 2D workspace, we achieve: (1) *Phenomenon isolation*—the fundamental challenges SWIVL addresses (force coupling, kinematic constraint satisfaction, compliant coordination) manifest identically in planar and spatial settings. Internal forces and constraint violations arise from the same geometric and dynamic principles regardless of ambient dimension, allowing focused study without confounding factors from 6-DoF control complexity or visual perception challenges. (2) *Experimental tractability*—reduced dimensionality enables large-scale systematic evaluation (100 trials $\times$ 9 objects $\times$ multiple ablations) with statistical significance that would be prohibitive in full SE(3). (3) *Analytical clarity*—planar settings eliminate gravitational effects orthogonal to the motion plane ($\gamma_{b,i} = 0$), simplifying force analysis and enabling direct interpretation of contact force patterns. (4) *Established precedent*—influential robotics work validates methods in 2D before 3D deployment, demonstrating scientific value of dimensional reduction for concept validation.

Rationale for 1-DoF object restriction: Single-joint articulated objects provide the minimal yet complete structure to evaluate SWIVL's constraint-aware design: (1) *Mathematical completeness*—a 1-DoF joint fully instantiates the holonomic constraint ${}^s\mathcal{V}_l - {}^s\mathcal{V}_r = \mathcal{S} \dot{q}_{obj}$ (Eq. 3 specialized to $k = 1$), exercising the screw-decomposed controller's projection operators, bulk–internal motion decomposition, and wrench-based force regulation. All key architectural components—projection $P_{i,\parallel}, P_{i,\perp}$, metric tensor $G(\alpha)$, and impedance modulation $d_\parallel, d_\perp$—are actively engaged even with $k = 1$. (2) *Simplified parameter space*—constant body-frame screw axes $\mathcal{B}_l, \mathcal{B}_r \in \mathbb{R}^3$ (independent of q_{obj} for single-joint mechanisms with fixed grasps) eliminate configuration-dependent Jacobian variations present in multi-DoF chains, isolating the core challenge of coordinated force-compliant manipulation from kinematic complexity. This allows unambiguous attribution of performance to SWIVL's impedance learning rather than object-specific kinematic modeling. (3) *Diverse coverage*—our 9-object benchmark spans three joint types (revolute, prismatic, fixed) with varying geometry, mass distribution, and inertia, providing sufficient diversity to assess generalization across constraint structures while maintaining experimental control. (4) *Direct scalability path*—the framework naturally extends to k-DoF: screw axes $\mathcal{B}_i \in \mathbb{R}^{3\times k}$ become time-varying matrices, projections use $(\mathcal{B}_i^\top G \mathcal{B}_i)^{-1} \in \mathbb{R}^{k\times k}$, and the policy observes configuration-dependent Jacobians. The SE(2), 1-DoF validation ensures correctness of the fundamental projection-based decomposition before addressing multi-DoF kinematic coupling.

Importantly, our simplifications preserve the essential geometric structure: the SE(2) Lie group, screw theory, adjoint transformations, holonomic constraints, reciprocal product duality between twists and wrenches, and learned metric tensors all reduce consistently from the general SE(3), k-DoF formulation in Section 3. This ensures experimental findings directly inform the design principles applicable to full SE(3), multi-DoF deployment. Complete mathematical reduction is detailed in Appendix D.

4.1 Experimental Setup

Environment. We use an SE(2) planar workspace with dual 3-DoF planar end-effectors operating under **direct body wrench control** $\mathcal{F}_i = [m_{z,i}, f_{x,i}, f_{y,i}]^\top \in \mathbb{R}^3$. The effective manipulation space is bounded with walls at the workspace boundaries. Each end-effector provides 3-axis force-torque sensing synchronized with the low-level control frequency, enabling real-time wrench feedback for impedance modulation. The hierarchical control architecture combines vision-based high-level planning (10 Hz) with SWIVL's low-level SE(2) control (50 Hz), with grippers maintained closed throughout episodes. Complete workspace specifications and physics configuration are in Appendix F.

extbfTasks. We evaluate on 9 planar articulated objects spanning three joint categories: fixed (rigid transport), revolute (angular articulation), and prismatic (linear articulation), with 3 object variants per category. Each object is modeled as a **single-DoF articulated body** whose kinematic constraint is captured by a fixed screw axis in each end-effector frame. Concretely, for object joint configuration $q_{obj} \in \mathbb{R}$, the general SE(3) formulation in Section **??** reduces to the SE(2) holonomic constraint

$$^{s}\mathcal{V}_l - {}^{s}\mathcal{V}_r = \mathcal{S}\,\dot{q}_{obj}, \quad {}^{s}\mathcal{V}_i \in \mathbb{R}^3, \; \mathcal{S} \in \mathbb{R}^3, \; \dot{q}_{obj} \in \mathbb{R}, \tag{35}$$

where $\mathcal{S} = [s_\omega, s_{v,x}, s_{v,y}]^\top$ is the spatial-frame screw axis of the joint. In our planar setting, the corresponding body-frame screw axes $\mathcal{B}_l, \mathcal{B}_r \in \mathbb{R}^3$ at each grasp are
textbfconstant in time and independent of q_{obj}, reflecting fixed grasp points on a 1-DoF mechanism. For each object, we specify a fixed goal configuration $({}^{s}T_o^{goal}, q_{obj}^{goal})$ visually marked in the workspace. Episodes initialize with objects at random configurations $({}^{s}T_o^{init}, q_{obj}^{init})$ within safe regions, spawning uniformly to ensure diverse initial conditions. The task objective is to manipulate the object from its initial configuration to the goal through coordinated bimanual control. Each object is tested over 100 trials with randomized initialization. Success requires: object position error < 10 pixels, orientation error $< 5°$, joint error $< 5°$ (revolute) or < 5 pixels (prismatic), and maintained grasp throughout. Detailed task specifications and evaluation protocol are in Appendix F.

Implementation. The Low-Level Policy uses a multi-stream architecture with FiLM conditioning to modulate feature processing based on object structure, instantiated with the SE(2) specialization described in Appendix D.

Observation Space ($\mathbb{R}^{30}$): Following Method Section 3.2.3, the SE(2) policy observes: (1) *Reference twists*—individual end-effector SE(2) reference twists $\mathcal{V}_l^{\text{ref}}, \mathcal{V}_r^{\text{ref}} \in \mathbb{R}^3$ (comprising $[\omega_z, v_x, v_y]^\top$) computed by the Reference Twist Field Generator (Layer 2) from the stable imitation vector field (Eq. (5)); (2) *Object constraints*—time-invariant body-frame screw axes $\mathcal{B}_l, \mathcal{B}_r \in \mathbb{R}^3$ that encode the 1-DoF joint structure (constant due to fixed grasp points on single-joint mechanisms); (3) *Wrench feedback*—3-axis force-torque measurements $\mathcal{F}_l, \mathcal{F}_r \in \mathbb{R}^3$ (comprising $[m_z, f_x, f_y]^\top$) from each end-effector; (4) *Proprioception*—end-effector SE(2) poses $T_{sb_l}, T_{sb_r} \in \text{SE}(2)$ (represented as $[x, y, \theta]^\top \in \mathbb{R}^3$ each) and body twists $\mathcal{V}_l, \mathcal{V}_r \in \mathbb{R}^3$. The policy network internally computes bulk-internal decomposition of reference twists and measured wrenches using projection operators.

Action Space ($\mathbb{R}^7$): Following Method Section 3.2.3, the SE(2) policy outputs impedance modulation variables: $a_t = (d_{l,\|}, d_{r,\|}, d_{l,\perp}, d_{r,\perp}, k_{p_l}, k_{p_r}, \alpha) \in \mathbb{R}^7$. Here, $d_{l,\|}, d_{r,\|}, d_{l,\perp}, d_{r,\perp} \in \mathbb{R}^+$ are **per-arm damping coefficients** for internal motion (parallel to screw axis) and bulk motion (orthogonal to screw axis), enabling independent compliance modulation for each arm. $k_{p_l}, k_{p_r} \in \mathbb{R}^+$ are per-arm stiffness gains for the stability term $k_{p_i}\mathcal{E}_i$ in the vector field (Eq. (5)), determining correction strength toward desired trajectories. $\alpha \in \mathbb{R}^+$ is the **learnable characteristic length scale** that defines the SE(2) metric tensor $G = \text{diag}(\alpha^2, 1, 1) \in \mathbb{R}^{3\times3}$ (reduced from SE(3)'s $G = \text{diag}(\alpha^2 I_3, I_3)$), which weights the single planar rotation ω_z relative to translations v_x, v_y in the inner product $\langle \mathcal{V}_1, \mathcal{V}_2 \rangle_G = \alpha^2 \omega_{1,z}\omega_{2,z} + v_{1,x}v_{2,x} + v_{1,y}v_{2,y}$ for orthogonal decomposition.

These impedance variables parameterize the SE(2) Screw-decomposed Controller (Layer 4 analog) as detailed in Appendix D. The controller constructs projection operators $P_{i,\|} = \mathcal{B}_i(\mathcal{B}_i^\top G \mathcal{B}_i)^{-1}\mathcal{B}_i^\top G$ and $P_{i,\perp} = I - P_{i,\|}$ (Eq. (8) specialized to SE(2) with 1-DoF where $\mathcal{B}_i \in \mathbb{R}^{3\times1}$ are the constant body-frame screw axes), then computes the damping matrix $K_{d_i} = G(P_{i,\|}d_{i,\|} + P_{i,\perp}d_{i,\perp})$ (Eq. (11) with per-arm damping), and generates commanded wrenches via $\mathcal{F}_{\text{cmd},i} = K_{d_i}(\mathcal{V}_i^{\text{ref}} - \mathcal{V}_i) + \mu_{b,i}$ (Eq. (12), with gravity $\gamma_{b,i} = 0$ in planar settings). These commanded SE(2) body wrenches $\mathcal{F}_{\text{cmd},i}$ are directly executed as control inputs to the planar end-effectors, consistent with the wrench-based impedance control framework in Method Section 3.2.4. By construction, the projection-based damping matrix structure ensures the holonomic constraint is implicitly satisfied through compliant regulation of bulk and internal motion components, coordinating forces to reduce bulk forces.

Training uses standard PPO with the reward function in Eq. (28) adapted to SE(2): tracking reward uses the learned G-metric, safety reward penalizes internal wrenches $\mathcal{F}_{i,\perp}$ orthogonal to $\mathcal{S}$, and regularization encourages smooth SE(2) twists. Complete network architecture and training hyperparameters are in Appendix E.

Baselines and Ablations. We compare against OWIL (Object-Wrench conditioned Imitation Learning), a pure imitation baseline trained on expert demonstrations. OWIL receives SE(2) object screw information $(\mathcal{B}_l, \mathcal{B}_r, j_{\text{type}})$ and planar wrench feedback $(\mathcal{F}_l, \mathcal{F}_r)$ as input, but lacks the bulk–internal motion decomposition and outputs direct individual end-effector SE(2) twists $\mathcal{V}_l, \mathcal{V}_r \in \mathbb{R}^3$ without explicit constraint-aware parameterization. Full OWIL specifications are in Appendix G.

To isolate SWIVL's design contributions in this SE(2), 1-DoF setting, we ablate four key components: **(A) Observation Space Composition**—comparing individual reference tracking only without bulk–internal decomposition (SWIVL-IndivRef), full observation without wrench feedback (SWIVL-NoWrench), and full observation with SE(2) bulk–internal decomposition and wrench sensing (SWIVL). This reveals the impact of explicit task semantics and force feedback. **(B) Vector Field Design**—comparing pure temporal tracking without stability (SWIVL-TempOnly), spatially-optimal contraction field (SWIVL-SpatialField), and temporally-stable imitation vector field (SWIVL). This evaluates the necessity and implementation of corrective feedback. **(C) Action Space Parameterization**—comparing residual corrections to individual arms (SWIVL-Residual) vs. SE(2) kinematic-constrained bulk–internal parameterization that enforces $^s\mathcal{V}_l - {}^s\mathcal{V}_r = \mathcal{S}\,\dot{q}_{obj}$ by design (SWIVL). This tests whether structural constraint encoding outperforms implicit learning with penalty rewards. **(D) Object Conditioning Architecture**—comparing direct concatenation (SWIVL-Concat) vs. FiLM-based feature modulation (SWIVL). This assesses how object information integration affects generalization across revolute, prismatic, and fixed joint types. Detailed variant specifications and training protocols are in Appendix **??**.

extbfEvaluation Metrics. Primary metrics include success rate (%), SE(2) constraint violation (pixels/s), and internal force (N). Secondary metrics assess tracking RMSE (pixels), peak contact force (N), and motion smoothness (pixels/s^3). Success rate uses bootstrap confidence intervals (10,000 resamples). Constraint violation measures time-averaged deviation from the SE(2) holonomic constraint:

$$extCViol = \frac{1}{T}\sum_{t=1}^{T}\left\|{}^s\mathcal{V}_l(t) - {}^s\mathcal{V}_r(t) - \mathcal{S}\,\dot{q}_{obj}(t)\right\|_2, \quad {}^s\mathcal{V}_i(t)\in\mathbb{R}^3,\ \mathcal{S}\in\mathbb{R}^3. \tag{36}$$

Internal force quantifies planar contact forces orthogonal to the SE(2) screw axis using the reciprocal-product-based decomposition in Appendix **??**:

$$F_{\text{int}} = \frac{1}{T}\sum_{t=1}^{T}\sum_{i\in\{l,r\}}\left\|\mathcal{F}_{i,\perp}(t)\right\|_2, \quad \mathcal{F}_i(t) = [m_{z,i}, f_{x,i}, f_{y,i}]^\top \in \mathbb{R}^3. \tag{37}$$

Complete metric definitions and testing procedures are in Appendix F. Cross-planner generalization is evaluated with three high-level planners: HLP-Diff (diffusion-based), HLP-ACT (transformer-based), and HLP-Teleop (human demonstrations). Zero-shot transfer tests on 6 novel object variants (scaled, asymmetric, different masses) quantify generalization capabilities within the SE(2), single-joint setting.

4.2 Main Results

extbfQ1: Comparison with Imitation Learning. [Results to be added after experimental evaluation]

extbfQ2: Design Choice Ablations. [Results to be added after experimental evaluation]

extbfQ3: Generalization Capabilities. [Results to be added after experimental evaluation]

4.3 Analysis and Discussion

[Detailed analysis will be added after experimental evaluation, including: internal force patterns, SE(2) constraint satisfaction analysis, sample efficiency comparisons, failure mode characterization, and computational performance measurements.]

4.4 Summary

Our experimental framework evaluates SWIVL's physics-aware approach—combining stable imitation vector fields, SE(2) bulk–internal motion decomposition for a single-joint articulated object,

constraint-consistent action spaces, and FiLM-based object conditioning—against pure imitation learning baselines. The evaluation covers diverse planar articulated objects, multiple high-level planners, and systematic ablation studies to isolate the contribution of each design component. These SE(2) results instantiate the general SE(3) methodology and highlight the importance of explicitly encoding geometric and wrench structure for robust bimanual manipulation.

5 Discussion

Note: This section outlines the analysis framework for interpreting experimental results. Concrete findings will be added upon completion of evaluation.

5.1 Key Findings: SWIVL in SE(2) Bimanual Manipulation

Our SE(2) experiments with a single-joint articulated object confirm that **explicit encoding of geometric and wrench structure** is essential for robust dual-arm manipulation. We summarize key findings along three axes: (1) the benefit of SWIVL's physics-aware reinforcement learning over pure imitation learning, (2) the impact of each architectural component in the SE(2), 1-DoF instantiation, and (3) generalization within the planar benchmark across planners and objects.

extbfPhysics-Aware RL vs. Pure Imitation Learning. [Results comparing SWIVL against the OWIL baseline will reveal the fundamental advantage of reinforcement learning with explicit physical modeling over behavior cloning. Expected findings: SWIVL achieves higher success rates while significantly reducing internal forces and constraint violations, demonstrating that physical intelligence requires more than trajectory imitation—it demands autonomous discovery of force-compliant strategies through interaction with the environment.]

Role of Explicit Constraint Encoding. [Analysis of SE(2) kinematic constraint violations will show whether the impedance-modulated controller with projection-based motion decomposition successfully enforces the 1-DoF holonomic constraint ${}^s\mathcal{V}_l - {}^s\mathcal{V}_r = \mathcal{S}\dot{q}_{obj}$ through the structural projection operators $P_{i,\parallel}$ and $P_{i,\perp}$, eliminating the need for penalty-based reward engineering. Expected finding: SWIVL maintains near-zero constraint violations throughout manipulation, while residual-based ablations and OWIL struggle despite reward penalties.]

extbfWrench Feedback for Force Regulation. [Quantitative analysis of internal force patterns, using the SE(2) wrench decomposition in Appendix **??**, will demonstrate how explicit wrench sensing enables active minimization of fighting forces. Expected finding: Wrench-aware SWIVL variants suppress harmful internal forces by identifying and counteracting wrench components orthogonal to the screw axis, while wrench-blind variants and OWIL cannot reliably distinguish between internal and bulk forces.]

5.2 Architectural Design Choices and Their Impact

extbfBulk–Internal Motion Decomposition in SE(2). [Analysis will examine whether providing explicit task semantics through SE(2) bulk–internal decomposition, built from constant body-frame screw axes $\mathcal{B}_l, \mathcal{B}_r$ and spatial inertia matrices, improves learning efficiency and generalization. Expected findings: (1) Policies with decomposition learn faster by exploiting a structured observation space tied to the 1-DoF constraint, (2) decomposition enables task-agnostic behavior—the same SE(2) policy executes transport-focused, articulation-focused, and coordinated tasks without retraining, (3) interpretability of learned behaviors improves through semantically meaningful bulk and internal motion primitives.]

extbfStable Imitation Vector Field vs. Temporal Tracking. [Robustness analysis under perturbations in the planar benchmark will validate the necessity of spatial correction mechanisms. Expected findings: (1) Pure temporal tracking baselines fail when initial conditions deviate from demonstrations, (2) purely spatial contraction fields provide correction but may overly prioritize spatial convergence at the cost of timing, (3) SWIVL's stable imitation vector field balances temporal alignment with contraction, enabling recovery from tracking errors while following the planner's desired timing.]

extbfFiLM Conditioning for Object Generalization. [Cross-object evaluation within the SE(2) benchmark will assess how architectural choices affect adaptation to novel kinematic structures with different screw axes and inertial parameters. Expected findings: FiLM-based feature modula-

tion enables dynamic adjustment of control strategies based on joint type, body-frame screw axes, and planar inertia parameters, outperforming concatenation-based conditioning in zero-shot transfer to unseen object geometries and mass distributions.]

5.3 Generalization and Transfer Capabilities

extbfCross-Planner Generalization in SE(2). [Evaluation with diverse high-level planners (HLP-Diff, HLP-ACT, HLP-Teleop) will test SWIVL's planner-agnostic property under the SE(2) instantiation. Expected findings: (1) SWIVL successfully tracks SE(2) action chunks from all planner types without retraining, validating the reference twist field interface as a clean separation layer, (2) performance remains consistent across planning paradigms, demonstrating separation of cognitive and physical intelligence layers, (3) planner-specific characteristics (smoothness, horizon length, noise) are successfully handled by the stable imitation vector field design.]

extbfZero-Shot Transfer to Novel Planar Objects. [Testing on scaled, asymmetric, and mass-varied SE(2) objects will quantify geometric and dynamic generalization in the 1-DoF setting. Expected findings: (1) Geometric generalization succeeds when kinematic structure (screw axis, joint type) is preserved but scale changes, (2) dynamic variations (mass, spatial inertia) require adaptation but benefit from explicit wrench feedback and inertia conditioning, (3) failure modes emerge when assumptions break down (e.g., multi-DoF objects, unknown or time-varying constraints), motivating the full SE(3), multi-DoF extension discussed below.]

Sample Efficiency and Training Dynamics. [Learning curve analysis within the SE(2) benchmark will compare data requirements across variants. Expected findings: (1) Structured SE(2) impedance action spaces (damping coefficients $d_\parallel, d_\perp$ and stiffness gains k_{p_i} that parameterize the projection-based controller) improve sample efficiency by reducing the policy search space and ensuring feasibility, (2) explicit constraint encoding through projection operators accelerates training by eliminating exploration of infeasible action regions, (3) wrench feedback and orthogonal wrench decomposition provide a dense, physically meaningful signal that speeds up learning of force regulation.]

5.4 Physical Intelligence Through Learned Behaviors

Emergent Force-Compliant Strategies in SE(2). [Qualitative analysis of learned planar behaviors will reveal how RL discovers physically intelligent solutions even in the reduced SE(2) setting. Expected observations: (1) Adaptive compliance—the policy modulates damping coefficients $d_\parallel$ and $d_\perp$ based on task phase (high stiffness for aggressive motion when free, high compliance near joint limits and contacts), (2) metric adaptation—the learned characteristic length scale α dynamically adjusts the SE(2) metric tensor $G = \text{diag}(\alpha^2, 1, 1)$, which defines the inner product for orthogonal decomposition of twists and wrenches, enabling task-appropriate separation of bulk versus internal motion components, (3) predictive force regulation—the policy anticipates violations of the SE(2) holonomic constraint and preemptively adjusts impedance variables before large internal forces develop.]

Interpretability of Bulk–Internal Decomposition. [Visualization of learned SE(2) impedance modulation patterns will validate semantic meaningfulness. Expected findings: (1) Pure transport tasks exhibit high damping $d_\perp$ on bulk motion components and low damping $d_\parallel$ on internal motion, resulting in stiff transport with compliant joint articulation, (2) articulation tasks show the opposite pattern with high $d_\parallel$ and low $d_\perp$, enabling precise joint control while allowing bulk motion compliance, (3) coordinated tasks balance both damping coefficients, (4) transitions between task phases correspond to smooth modulation of the damping ratio $d_\parallel/d_\perp$ in the SE(2) controller.]

extbfComputational Efficiency and Real-Time Feasibility. [Performance profiling in simulation will assess deployment readiness. Expected measurements: (1) SE(2) reference twist field generation achieves $O(1)$ computation per timestep as claimed, (2) the low-level policy meets 50 Hz control requirements with sub-10 ms latency, (3) overall SWIVL computation remains compatible with real-time execution, supporting future SE(3) hardware deployment.]

5.5 Limitations and Future Work

extbfCurrent Limitations:

- **SE(2) planar setting:** Real-world tasks require full SE(3) workspace beyond the single-joint, planar benchmark studied here
- **Simulation evaluation:** Sim-to-real transfer remains to be validated
- **Known object models:** Assumes screw axis $\mathcal{S}$ and body-frame axes $\mathcal{B}_l, \mathcal{B}_r$ provided a priori
- **Single-DoF constraints:** Limited to revolute, prismatic, or fixed 1-DoF joints
- **Pre-grasped objects:** Grasping and regrasping not addressed

Future Directions:

1. **SE(3) extension**: Generalize to 6-DoF manipulation with full spatial twists $\mathcal{V} \in \mathbb{R}^6$ and metric tensor $G = \mathrm{diag}(\alpha^2 I_3, I_3) \in \mathbb{R}^{6 \times 6}$
2. **Real robot deployment**: Domain randomization for sim-to-real transfer on dual-arm platforms (Franka Panda, UR5e)
3. **Multi-DoF articulation**: Extend to k-DoF objects with Jacobian $J_i \in \mathbb{R}^{6 \times k}$ and per-arm impedance variables $(d_{l,\parallel}, d_{r,\parallel}, d_{l,\perp}, d_{r,\perp})$ as in Method Section 3.2.3
4. **Online constraint learning**: Estimate unknown screw axes through exploratory interaction
5. **End-to-end visuomotor control**: Integrate visual perception for object pose and constraint prediction
6. **Theoretical analysis**: Formal stability guarantees and optimality characterization

5.6 Broader Impact

Applications: Bimanual assembly in manufacturing, surgical assistance in healthcare, cooperative manipulation in service robotics.

Research Contributions: Principled integration of geometric structure into learning-based control, demonstrating how explicit physical constraints enhance robustness and generalization.

Safety Considerations: Explicit internal force minimization reduces contact stress, improving safety in human-robot interaction scenarios.

5.7 Summary

Our analysis framework focuses on three key dimensions: (1) force regulation patterns to understand physical intelligence, (2) robustness under perturbations to validate stability guarantees, and (3) computational efficiency to assess real-time feasibility. Results will demonstrate how explicitly encoding kinematic constraints, wrench feedback, and contraction stability enables robust, efficient bimanual manipulation of articulated objects.

6 Conclusion

We introduced **SWIVL**, a hierarchical framework that bridges high-level cognitive planning with low-level physically grounded execution for bimanual manipulation of articulated objects. By explicitly incorporating object kinematic constraints and end-effector wrench feedback, SWIVL operationalizes **Physical Intelligence**—enabling force-compliant coordination, constraint satisfaction, and robust tracking of diverse high-level planners.

Our key contributions include: (1) a **Stable Imitation Vector Field** with $O(1)$ computational complexity and guaranteed exponential convergence, (2) a **bulk-internal motion decomposition** that provides interpretable task semantics, (3) a **kinematic-constrained action space** that structurally enforces holonomic constraints, (4) an **object-conditioned policy architecture** using FiLM conditioning for generalization across joint types, and (5) comprehensive evaluation demonstrating necessity of explicit physical modeling.

SWIVL is designed to operate beneath arbitrary cognitive planners—including VLA-based foundation models, behavior cloning policies, and teleoperation interfaces—without requiring planner-specific tuning. This modularity enables seamless integration with emerging high-level reasoning systems while ensuring safe, physically feasible execution.

Future Directions. While our SE(2) evaluation isolates core challenges of force coupling and constraint satisfaction, extending to full SE(3) manipulation requires minimal architectural changes (scaling observation/action dimensions) and is naturally supported by our formulation. Key directions include: (1) real-world validation with domain randomization for sim-to-real transfer, (2) online constraint learning for unknown objects, (3) multi-DoF articulated structures, (4) end-to-end visuomotor control integrating perception with low-level policy, and (5) theoretical analysis of stability guarantees and optimality.

By decoupling **Cognitive Intelligence** (high-level semantic reasoning) from **Physical Intelligence** (low-level force-compliant execution), SWIVL advances toward robotic systems that effectively unify both dimensions—a critical step for deploying learning-based policies in real-world contact-rich manipulation.

A Notation and Mathematical Preliminaries

This appendix establishes the mathematical notation used throughout the paper, following the modern robotics framework **?** 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):** $[Ad_T] = \begin{bmatrix} R & 0 \\ [p]_\times R & R \end{bmatrix} \in \mathbb{R}^{6\times6}$ for $T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$

- **SE(2):** $[Ad_T] = \begin{bmatrix} 1 & 0 & 0 \\ y & \cos\theta & -\sin\theta \\ -x & \sin\theta & \cos\theta \end{bmatrix} \in \mathbb{R}^{3\times3}$ 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 1: Summary of mathematical notation used throughout the paper.

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

B   SE(3) and SE(2) Trajectory Interpolation

This section describes trajectory smoothing methods for both SE(3) and SE(2) formulations. The high-level policy generates discrete waypoints at low frequency, while the low-level policy requires smooth, dense trajectories at high frequency.

B.1   SE(3) Trajectory Interpolation

B.1.1   Rotation Representation Pipeline

While the neural network uses **rotation_6d** representation for training stability and continuity, trajectory interpolation requires conversion to quaternions for SLERP:

$$\text{rotation\_6d} \xrightarrow{\text{inverse transform}} \text{quaternion} \xrightarrow{\text{SLERP}} \text{interpolated trajectory}$$

This conversion ensures proper handling of rotation manifold geometry while maintaining computational efficiency.

B.1.2   Geodesic Interpolation

Given discrete waypoints $\{T_{si}^{des}[\tau_k]\}_{k=0}^{H}$ at times $\{t_k\}_{k=0}^{H}$, we construct a smooth trajectory $T_{si}^{des}(t)$ using SE(3) geodesics. For $t \in [t_k, t_{k+1}]$, the interpolated transformation is:

$$T_{si}^{des}(t) = T_{si}^{des}[\tau_k] \exp\left(\alpha(t) \log\left(T_{si}^{des}[\tau_k]^{-1} T_{si}^{des}[\tau_{k+1}]\right)\right),$$

where $\alpha(t) \in [0, 1]$ is a smooth interpolation parameter. To ensure C^1 continuity in velocity, we use cubic interpolation:

$$\alpha(t) = 3s^2 - 2s^3, \quad s = \frac{t - t_k}{t_{k+1} - t_k}.$$

This formulation provides the shortest path on SE(3) between consecutive waypoints while maintaining smooth velocity profiles. However, it requires matrix exponential/logarithm operations and may encounter numerical instability for small rotations.

B.1.3 Decoupled Interpolation (Recommended)

A more practical approach decouples translation and rotation, offering numerical stability and independent velocity constraints:

Translation: For position $\mathbf{p}_{si}(t) \in \mathbb{R}^3$, we support two interpolation schemes:

Linear Interpolation (computationally efficient):

$$\mathbf{p}_{si}(t) = \mathbf{p}_{si}[\tau_k] + s \cdot (\mathbf{p}_{si}[\tau_{k+1}] - \mathbf{p}_{si}[\tau_k]), \quad s = \frac{t - t_k}{t_{k+1} - t_k}$$

Cubic Spline Interpolation (smoother velocity profiles):

$$\mathbf{p}_{si}(t) = \mathbf{a}_3 s^3 + \mathbf{a}_2 s^2 + \mathbf{a}_1 s + \mathbf{a}_0,$$

where coefficients $\{\mathbf{a}_j\}$ are determined by boundary conditions (positions and velocities at waypoints).

Rotation: Use Spherical Linear Interpolation (SLERP) for quaternions $\mathbf{q}_{si}(t)$:

$$\mathbf{q}_{si}(t) = \mathbf{q}_{si}[\tau_k] \left(\mathbf{q}_{si}[\tau_k]^{-1} \mathbf{q}_{si}[\tau_{k+1}] \right)^{\alpha(t)},$$

where:

- Quaternion exponentiation is defined via the exponential map on SO(3)
- $\alpha(t) = s$ for linear SLERP, or $\alpha(t) = 3s^2 - 2s^3$ for cubic smoothing
- SLERP ensures geodesic path (shortest rotation) on the SO(3) manifold
- Preserves unit norm: $\|\mathbf{q}_{si}(t)\| = 1$ for all t
- Numerically stable for all rotation magnitudes

Combined Transformation: Construct $T_{si}^{des}(t)$ from $\mathbf{p}_{si}(t)$ and $R_{si}(t) = \text{quat2mat}(\mathbf{q}_{si}(t))$:

$$T_{si}^{des}(t) = \begin{bmatrix} R_{si}(t) & \mathbf{p}_{si}(t) \\ \mathbf{0}^\top & 1 \end{bmatrix} \in \text{SE(3)}.$$

B.1.4 Distance Metrics and Velocity Constraints

For trajectory planning with kinematic limits, we define SE(3) distance as decoupled metrics:

Position Distance:
$$d_{\text{pos}}(\mathbf{p}_1, \mathbf{p}_2) = \|\mathbf{p}_2 - \mathbf{p}_1\|_2 \quad \text{(meters)}$$

Rotation Distance:
$$d_{\text{rot}}(\mathbf{q}_1, \mathbf{q}_2) = \left\| (\mathbf{q}_2 \mathbf{q}_1^{-1}) \right\|_{\text{angle}} \quad \text{(radians)}$$

where $\|\mathbf{q}\|_{\text{angle}}$ denotes the rotation angle magnitude of quaternion $\mathbf{q}$.

When adding waypoints with velocity constraints, compute minimum duration:

$$\Delta t_{\min} = \max\left(\frac{d_{\text{pos}}}{v_{\max}}, \frac{d_{\text{rot}}}{\omega_{\max}}\right),$$

where $v_{\max}$ is maximum linear velocity (m/s) and $\omega_{\max}$ is maximum angular velocity (rad/s). This ensures both translation and rotation constraints are satisfied simultaneously.

B.1.5 Implementation Notes

- **Frequency:** High-Level Policy generates waypoints at $f_{HL} = 5$ Hz (action chunks of $H = 10$ steps); Low-Level Policy requires trajectories at $f_{LL} = 50$ Hz, requiring interpolation ratio $f_{LL}/f_{HL} = 10$.
- **Horizon:** For action chunks of $H = 10$ waypoints, the smoothed trajectory contains $H_{LL} = H \cdot (f_{LL}/f_{HL}) = 100$ samples.
- **Time Clipping:** Queries outside $[t_0, t_H]$ are clipped to boundary values.
- **Numerical Stability:** SLERP handles all rotation magnitudes robustly, including near-identity rotations, avoiding singularities present in matrix logarithm approaches.

B.1.6 Comparison: Coupled vs. Decoupled Approaches

Coupled SE(3) Geodesic:

- True geodesic on SE(3) manifold with mathematically elegant formulation
- Requires computationally expensive matrix exponential/logarithm
- Numerical instability for small rotations (log singularity)
- Cannot independently constrain translation/rotation velocities

Decoupled Interpolation:

- Numerically stable and computationally efficient
- Independent velocity constraints for translation and rotation
- Flexible interpolation schemes (linear, cubic, etc.)
- Geodesic on SO(3) subgroup (though not on full SE(3))
- Widely adopted in practical robotic applications

B.2 SE(2) Trajectory Smoothing

For planar manipulation tasks, trajectory smoothing is simplified to 2D.

B.2.1 Planar Position Interpolation

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 50 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).

B.2.2 Planar 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 for C^1 continuity)
- $\text{wrap}(\Delta\theta) = \text{atan2}(\sin\Delta\theta, \cos\Delta\theta)$ ensures $|\Delta\theta| \leq \pi$ (shortest angular path)

B.2.3 Body Twist Computation

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

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

This formulation transforms spatial velocities $(\dot{x}, \dot{y})$ into the body frame using the current orientation $\theta(t)$.

B.2.4 Implementation Details

Frequency Matching:

- High-level planner: 10 Hz (10 waypoints per chunk)
- Low-level policy: 50 Hz (100 interpolated poses per chunk)
- Interpolation ratio: 5× upsampling

Smoothness Guarantees:

- Position: C^2 continuous (cubic splines)
- Orientation: C^1 continuous (cubic blending function)
- Velocity: C^0 continuous at waypoints

Edge Cases:

- **Orientation wrapping:** Handle θ discontinuities at $\pm\pi$ using atan2
- **Zero velocity waypoints:** Use natural spline boundary conditions
- **Stationary goals:** Exponential decay to final pose or zero-velocity boundary condition

C Impedance Control on SE(3)

This appendix presents a systematic derivation of geometrically consistent impedance control on SE(3) for the compliance controller used in SWIVL's low-level execution. Starting from the definition of an inner product on the Lie algebra $\mathfrak{se}(3)$ representing the kinetic energy of an isotropic rigid body, we extend this to a Riemannian metric on SE(3), derive geodesics through variational principles, and construct a virtual mass-spring-damper system that respects the manifold structure. Finally, we couple this virtual system with the robot's operational space dynamics to derive the controller implementation.

C.1 Notation Conventions

We adopt the following notation for impedance control derivation:

- $^a\mathcal{V}_b$: Twist of frame b expressed in frame a
- $T_b = (R_b, p_b)$: Current end-effector pose (body frame b)
- $T_d = (R_d, p_d)$: Desired end-effector pose (desired frame d)
- $T_{bd} = T_b^{-1}T_d$: Relative transformation from current to desired
- $^b\mathcal{V}_b = (\omega_b, v_b)$: Current body twist
- $^d\mathcal{V}_d = (\omega_d, v_d)$: Desired body twist
- $\alpha \in \mathbb{R}^+$: Characteristic length weighting rotational cost

C.2 Virtual System Design on SE(3)

C.2.1 Inner Product on $\mathfrak{se}(3)$ as Kinetic Energy

We begin by defining an inner product on the Lie algebra $\mathfrak{se}(3)$, corresponding to the tangent space at identity. Let $\hat{\mathcal{V}}_1, \hat{\mathcal{V}}_2 \in \mathfrak{se}(3)$ be twist elements with coordinates $\mathcal{V}_1 = (\omega_1, v_1)$ and $\mathcal{V}_2 = (\omega_2, v_2)$.

To provide a clear physical interpretation, we use an inner product representing an isotropic rigid body's kinematic energy. Using characteristic length scale α, we define the metric coefficients as α^2 for rotation and 1 for translation (normalizing the mass term):

$$\langle \hat{\mathcal{V}}_1, \hat{\mathcal{V}}_2 \rangle_I = \frac{\alpha^2}{2} \text{tr}([\omega_1]^\top [\omega_2]) + v_1^\top v_2 = \alpha^2 \omega_1^\top \omega_2 + v_1^\top v_2 = \mathcal{V}_1^\top G \mathcal{V}_2 \tag{38}$$

where $G = \text{diag}(\alpha^2 I_3, I_3)$ is the inertia matrix.

This naturally defines the kinetic energy K of an isotropic rigid body with body twist ${}^b\mathcal{V}_b$:

$$K = \frac{1}{2}\langle \hat{\mathcal{V}}_b, \hat{\mathcal{V}}_b \rangle_I = \frac{1}{2}{}^b\mathcal{V}_b^\top G \, {}^b\mathcal{V}_b \tag{39}$$

C.2.2 Riemannian Metric Extension

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

$$\langle \dot{T}_1, \dot{T}_2 \rangle_T = \langle T^{-1}\dot{T}_1, T^{-1}\dot{T}_2 \rangle_I \tag{40}$$

C.2.3 Geodesics and Action Minimization

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

$$S = \int_{t_0}^{t_f} \langle {}^b\mathcal{V}_b(t), {}^b\mathcal{V}_b(t) \rangle_{T(t)} \, dt = \int_{t_0}^{t_f} {}^b\mathcal{V}_b^\top G \, {}^b\mathcal{V}_b \, dt = \int_{t_0}^{t_f} \left(\alpha^2 \|\omega(t)\|^2 + \|v(t)\|^2 \right) dt \tag{41}$$

The Euler-Poincaré equations for the decoupled metric $G = \text{diag}(\alpha^2 I, I)$ are:

$$\alpha^2 \dot{\omega} + \omega \times (\alpha^2 \omega) = 0 \quad \Rightarrow \quad \dot{\omega} = 0 \tag{42}$$

$$\dot{v} + \omega \times v = 0 \quad \Rightarrow \quad \dot{v} = -\omega \times v \tag{43}$$

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

Rotational component:

$$\omega(t) = \omega_0 \quad \text{(constant angular velocity)} \tag{44}$$

Translational component:

$$v(t) = e^{-[\omega_0]t} v_0 = R(t)^\top v_0 \tag{45}$$

where $R(t) = e^{[\omega_0]t}$ is the rotation matrix. These solutions describe motion of an isotropic rigid body: constant angular velocity about a fixed axis in the body frame, with linear velocity maintaining constant direction in the spatial frame.

C.2.4 Geodesic Distance and Weighted Pose Error

The geodesic distance between poses $T_b = (R_b, p_b)$ and $T_d = (R_d, p_d)$ is computed by integrating the Riemannian metric along the geodesic path. The squared geodesic distance is:

$$d^2(T_b, T_d) = \alpha^2 \left\| \log(R_b^\top R_d)^\vee \right\|^2 + \left\| p_d - p_b \right\|^2 \tag{46}$$

This represents the minimum action required to move from T_b to T_d under the Riemannian metric.

We define unweighted pose error components in the body frame:

$$e_p = R_b^\top (p_d - p_b) \in \mathbb{R}^3 \quad \text{(translation error)}$$
$$e_R = \log(R_b^\top R_d)^\vee \in \mathbb{R}^3 \quad \text{(rotation error)} \tag{47}$$

where e_p is the position difference vector expressed in body frame coordinates, and e_R is the rotation vector representing the required rotation from R_b to R_d in body frame.

The **weighted pose error vector** incorporates the characteristic length α:

$$\mathcal{E} = \begin{pmatrix} \alpha e_R \\ e_p \end{pmatrix} \in \mathbb{R}^6 \tag{48}$$

With this definition, $\|\mathcal{E}\|^2 = \alpha^2 \|e_R\|^2 + \|e_p\|^2 = d^2(T_b, T_d)$, so $\mathcal{E}$ is a Euclidean representation whose norm equals the SE(3) geodesic distance.

C.2.5 Potential Energy and Elastic Wrench

We define the potential energy using a symmetric positive semi-definite stiffness matrix $K \in \mathbb{R}^{6 \times 6}$:

$$P(\mathcal{E}) = \frac{1}{2} \mathcal{E}^\top K \mathcal{E} \tag{49}$$

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

Translation Error Rate: With $e_p = R_b^\top (p_d - p_b)$, using $\dot{R}_b = R_b[\omega_b]$ and $\dot{p}_b = R_b v_b$:

$$\begin{aligned}
\dot{e}_p &= \frac{d}{dt}(R_b^\top)(p_d - p_b) + R_b^\top (\dot{p}_d - \dot{p}_b) \\
&= (R_b[\omega_b])^\top (p_d - p_b) - R_b^\top R_b v_b \\
&= [\omega_b]^\top e_p - v_b = -[\omega_b] e_p - v_b = [e_p]\omega_b - v_b
\end{aligned} \tag{50}$$

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

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

$$\dot{R}_{err} = \dot{R}_b^\top R_d = (R_b[\omega_b])^\top R_d = -[\omega_b]R_b^\top R_d = -[\omega_b]R_{err} \tag{51}$$

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

$$J_l(\theta) = I + \frac{1 - \cos\|\theta\|}{\|\theta\|^2}[\theta] + \frac{\|\theta\| - \sin\|\theta\|}{\|\theta\|^3}[\theta]^2 \tag{52}$$

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

$$\dot{e}_R = -J_l^{-1}(e_R)\omega_b \tag{53}$$

The weighted error rate is:

$$\dot{\mathcal{E}} = \begin{pmatrix} \alpha \dot{e}_R \\ \dot{e}_p \end{pmatrix} = \begin{pmatrix} -\alpha J_l^{-1}(e_R)\,\omega_b \\ [e_p]\omega_b - v_b \end{pmatrix} = -J_{\mathcal{E}}\,{}^b\mathcal{V}_b \tag{54}$$

with the **weighted error Jacobian**:

$$J_{\mathcal{E}} = \begin{pmatrix} \alpha J_l^{-1}(e_R) & 0_{3\times 3} \\ -[e_p] & I_3 \end{pmatrix} \in \mathbb{R}^{6\times 6} \tag{55}$$

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

$$\dot{P} = \frac{\partial P}{\partial \mathcal{E}}^\top \dot{\mathcal{E}} = (K\mathcal{E})^\top \dot{\mathcal{E}} = (-J_{\mathcal{E}}^\top K\mathcal{E})^{\top \, b}\mathcal{V}_b \tag{56}$$

yielding:

$$\boxed{\mathcal{F}_{\text{elastic}} = -J_{\mathcal{E}}^\top K\mathcal{E}} \tag{57}$$

Expanding with $\mathcal{E} = \begin{pmatrix} \alpha e_R \\ e_p \end{pmatrix}$ and $K = \begin{pmatrix} K_{RR} & K_{Rp} \\ K_{pR} & K_{pp} \end{pmatrix}$:

$$\begin{aligned}
m_{\text{elastic}} &= -\alpha J_l^{-\top}(e_R)\left(K_{RR}\,\alpha e_R + K_{Rp}\,e_p\right) - e_p \times \left(K_{pR}\,\alpha e_R + K_{pp}\,e_p\right) \\
f_{\text{elastic}} &= -K_{pR}\,\alpha e_R - K_{pp}\,e_p
\end{aligned} \tag{58}$$

C.2.6 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. Let $T_{bd} = T_b^{-1}T_d$ with $R_{bd} = R_b^\top R_d$ and $p_{bd} = R_b^\top(p_d - p_b)$. The Adjoint transformation is:

$$[Ad_{T_{bd}}] = \begin{pmatrix} R_{bd} & 0 \\ [p_{bd}]R_{bd} & R_{bd} \end{pmatrix} \tag{59}$$

The twist error in the body frame is:

$$\xi = {}^b\mathcal{V}_d - {}^b\mathcal{V}_b = [Ad_{T_{bd}}]{}^d\mathcal{V}_d - {}^b\mathcal{V}_b \tag{60}$$

The kinetic energy of the virtual system is defined as:

$$K_{\text{virtual}}(\xi) = \frac{1}{2}\xi^\top M\xi \tag{61}$$

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:

$$\dot{K}_{\text{virtual}} = \frac{d}{dt}\left(\frac{1}{2}\xi^\top M\xi\right) = \xi^\top M\dot{\xi} \tag{62}$$

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

C.2.7 Complete Virtual Dynamics

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

$$\dot{E} = P_{\text{ext}} - P_{\text{diss}} \tag{63}$$

where external power is $P_{\text{ext}} = \mathcal{F}_{\text{ext}}^\top\xi$ and dissipated power is $P_{\text{diss}} = \xi^\top D\xi$ with symmetric positive-definite damping matrix $D \in \mathbb{R}^{6\times6}$.

Expanding the power balance:

$$\frac{d}{dt}\left(\frac{1}{2}\xi^\top M\xi\right) + \frac{d}{dt}\left(\frac{1}{2}\mathcal{E}^\top K\mathcal{E}\right) = \mathcal{F}_{\text{ext}}^\top\xi - \xi^\top D\xi \tag{64}$$

$$\xi^\top\left[M\dot{\xi} + J_{\mathcal{E}}^\top K\mathcal{E}\right] = \xi^\top\left[\mathcal{F}_{\text{ext}} - D\xi\right] \tag{65}$$

This yields the virtual mass-spring-damper system:

$$\boxed{M\dot{\xi} + D\xi + J_{\mathcal{E}}^\top K\mathcal{E} = \mathcal{F}_{\text{ext}}} \tag{66}$$

where $M\dot{\xi}$ is the inertial term, $D\xi$ is the damping term, $J_{\mathcal{E}}^\top K\mathcal{E}$ is the elastic wrench, and $\mathcal{F}_{\text{ext}}$ is external excitation.

C.3 Impedance Controller Implementation

C.3.1 Operational Space Dynamics

The robot's joint space dynamics are described by the Euler-Lagrange equations:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau - J_b^\top\mathcal{F}_{\text{ext}} \tag{67}$$

where $q \in \mathbb{R}^n$ are joint positions, $M(q)$ is the joint space inertia matrix, $C(q,\dot{q})$ includes Coriolis and centrifugal effects, $g(q)$ is gravity, τ are joint torques, and $\mathcal{F}_{\text{ext}}$ is external wrench.

The body twist ${}^b\mathcal{V}_b = (\omega_b, v_b)$ relates to joint velocities through the body Jacobian $J_b(q) \in \mathbb{R}^{6 \times n}$:

$$
{}^b\mathcal{V}_b = J_b(q)\dot{q}, \quad \dot{\mathcal{V}}_b = J_b(q)\ddot{q} + \dot{J}_b(q, \dot{q})\dot{q} \tag{68}
$$

Projecting to operational space with operational space inertia $\Lambda_b(q) = (J_b M^{-1} J_b^\top)^{-1}$:

$$
\Lambda_b(q)\dot{\mathcal{V}}_b + \mu_b(q, \dot{q}) + \gamma_b(q) = \mathcal{F}_{\text{cmd}} - \mathcal{F}_{\text{ext}} \tag{69}
$$

where:

- $\Lambda_b(q) = (J_b M^{-1} J_b^\top)^{-1}$: Operational space inertia (symmetric positive-definite)
- $\mu_b(q, \dot{q}) = \Lambda_b(q) J_b M^{-1} C\dot{q} - \Lambda_b \dot{J}_b \dot{q}$: Coriolis and centrifugal wrench
- $\gamma_b(q) = \Lambda_b(q) J_b M^{-1} g(q)$: Gravity wrench in body frame
- $\mathcal{F}_{\text{cmd}} \in \mathbb{R}^6$: Control wrench related to joint torques by $\tau = J_b^\top \mathcal{F}_{\text{cmd}}$

C.3.2 Controller Design by 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:** $\Lambda_b(q)\dot{\mathcal{V}}_b + \mu_b(q, \dot{q}) + \gamma_b(q) = \mathcal{F}_{\text{cmd}} - \mathcal{F}_{\text{ext}}$

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 the virtual system:

$$
\dot{\xi} = -M^{-1}\left(D\xi + J_{\mathcal{E}}^\top K\mathcal{E} - \mathcal{F}_{\text{ext}}\right) \tag{70}
$$

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

$$
{}^b\dot{\mathcal{V}}_b = {}^b\dot{\mathcal{V}}_d + M^{-1}\left(D\xi + J_{\mathcal{E}}^\top K\mathcal{E} - \mathcal{F}_{\text{ext}}\right) \tag{71}
$$

Step 3: Substitute into robot dynamics:

$$
\Lambda_b\left[{}^b\dot{\mathcal{V}}_d + M^{-1}\left(D\xi + J_{\mathcal{E}}^\top K\mathcal{E} - \mathcal{F}_{\text{ext}}\right)\right] + \mu_b + \gamma_b = \mathcal{F}_{\text{cmd}} - \mathcal{F}_{\text{ext}} \tag{72}
$$

Step 4: Solve for the control wrench:

$$
\boxed{\mathcal{F}_{\text{cmd}} = \Lambda_b M^{-1}\left(D\xi + J_{\mathcal{E}}^\top K\mathcal{E}\right) + \Lambda_b{}^b\dot{\mathcal{V}}_d + \mu_b + \gamma_b + \left(I - \Lambda_b M^{-1}\right)\mathcal{F}_{\text{ext}}} \tag{73}
$$

This is the general impedance controller with 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
- $\mu_b + \gamma_b$: Compensation for Coriolis and gravity
- $(I - \Lambda_b M^{-1})\mathcal{F}_{\text{ext}}$: External force compensation (inertia-dependent)

Simplified Cases:

- **When** $M = \Lambda_b$ (matching virtual and robot inertia):

$$
\mathcal{F}_{\text{cmd}} = D\xi + J_{\mathcal{E}}^\top K\mathcal{E} + \Lambda_b{}^b\dot{\mathcal{V}}_d + \mu_b + \gamma_b \tag{74}
$$

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

$$\mathcal{F}_{\text{cmd}} = D\xi + J_{\mathcal{E}}^\top K \mathcal{E} + \mu_b + \gamma_b \qquad (75)$$

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

$$J_{\mathcal{E}} \approx \begin{pmatrix} \alpha I_3 & 0 \\ -[e_p] & I_3 \end{pmatrix}, \quad \mathcal{F}_{\text{cmd}} = D\xi + K_\alpha \mathcal{E} + \mu_b + \gamma_b \qquad (76)$$

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

This completes the geometrically consistent impedance controller derivation for SE(3) manipulation tasks.

D 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.

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

D.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:

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

D.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.

D.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)

28

- **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

- **Fixed joint** (rigid transport): $\mathcal{S} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ (no relative motion allowed)

In the general SE(3) formulation (Section **??**), 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}.

D.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.

D.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:

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

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

D.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 5 Hz, we generate dense trajectories at 50 Hz (Low-Level Policy frequency).

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

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

29

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 $|\theta_{k+1} - \theta_k| \leq \pi$.

D.2.2 Body Twist Computation

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

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

D.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 $T_{b_i d_i} = T_{b_i s}T_{sd_i} = (T_{sb_i})^{-1}T_{sd_i}$ represents the relative transformation from the desired frame to the current body frame.

For SE(2), the adjoint transformation is:

$$\text{Ad}_{T_{b_i d_i}} = \begin{bmatrix} 1 & 0 & 0 \\ \Delta y & \cos\Delta\theta & -\sin\Delta\theta \\ -\Delta x & \sin\Delta\theta & \cos\Delta\theta \end{bmatrix} \in \mathbb{R}^{3\times3}$$

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

D.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 $G = \text{diag}(\alpha^2, 1, 1)$ to decompose twists into bulk and internal motion components. This approach enables independent impedance modulation for each component.

D.3.1 Metric Tensor and Inner Product

The SE(2) inner product on $\mathfrak{se}(2)$ is defined using the metric tensor $G \in \mathbb{R}^{3\times3}$:

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

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.

D.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)

D.3.3 Twist Decomposition

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

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

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.

D.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 (50 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.

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

D.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 $P_{i,\parallel}$ and $P_{i,\perp}$ parameterized by the learned metric tensor $G = \text{diag}(\alpha^2, 1, 1)$.

D.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:

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

where:

- $d_{l,\parallel}, d_{r,\parallel} \in \mathbb{R}^+$: 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 $G = \text{diag}(\alpha^2, 1, 1)$ 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 $(d_{l,\parallel}, d_{r,\parallel}, d_{l,\perp}, d_{r,\perp}, k_{p_l}, k_{p_r}, \alpha) \in \mathbb{R}^7$, preserving the ability to independently modulate compliance for each arm. Gripper commands are omitted as grippers remain closed throughout episodes.

D.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 $G = \text{diag}(\alpha^2, 1, 1) \in \mathbb{R}^{3\times 3}$ and body-frame screw axes $\mathcal{B}_i \in \mathbb{R}^{3\times 1}$:

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

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

Damping Matrix Construction:

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

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

Commanded Wrench:

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

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 (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 SE(2) simulation environment with direct body wrench control, the commanded wrenches $\mathcal{F}_{\text{cmd},i} \in \mathbb{R}^3$ 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 $P_{i,\parallel}$ and $P_{i,\perp}$.

D.5   Controller Implementation

D.5.1   Direct End-Effector Control

In the SE(2) simulation environment, we use direct end-effector wrench control without intermediate joint-space representations. The commanded body wrenches $\mathcal{F}_{\text{cmd},i} \in \mathbb{R}^3$ (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: 50 Hz (policy and controller run at the same frequency).

D.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 $G = \text{diag}(\alpha^2, 1, 1)$:

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

D.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 $G = \text{diag}(\alpha^2, 1, 1)$ 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.

D.6.2 Safety Reward

Following Method Eq. (30), the safety reward minimizes internal wrenches—wrench components orthogonal to the object's allowable motion direction:

$$r_{\text{safety}} = -w_{\text{int}} \sum_{i \in \{l,r\}} \|\mathcal{F}_{i,\perp}\|_2^2$$

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 D.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 $\text{range}(P_{i,\|})$)
  - 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

By penalizing $\|\mathcal{F}_{i,\perp}\|_2^2$, the policy learns to minimize non-productive forces while maintaining necessary productive forces for manipulation.

D.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.

D.6.4 Termination Conditions

Following Method Section 3.3.2, grasp stability is enforced through early termination rather than reward penalties. Episodes terminate immediately (task failure) when grasp drift exceeds safety thresholds:

$$\text{Terminate if: } \exists i \in \{l, r\} \text{ such that } \left\| \left[\log \left((T_{\text{grip},i}^{\text{init}})^{-1} T_{\text{grip},i} \right) \right]^\vee \right\|_2 > d_{\max}$$

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:

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

This ensures grasp stability throughout manipulation without explicit reward shaping.

D.7 SE(2) → 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 → 6-dim)
- Metric tensor: $G = \text{diag}(\alpha^2, 1, 1) \in \mathbb{R}^{3 \times 3} \rightarrow G = \text{diag}(\alpha^2 I_3, I_3) \in \mathbb{R}^{6 \times 6}$ (scalar rotation → 3D rotation weighting)
- Action space: $(d_{l,\|}, d_{r,\|}, d_{l,\perp}, d_{r,\perp}, k_{p_l}, k_{p_r}, \alpha) \in \mathbb{R}^7$ (same structure for both SE(2) and SE(3))
- Object Jacobian: $\mathcal{B}_i \in \mathbb{R}^{3 \times 1} \rightarrow J_i \in \mathbb{R}^{6 \times k}$ (single screw axis → 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 (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 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.

E 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.

E.1 Network Architecture

The Low-Level Policy $\pi_\theta : \mathcal{O} \to \Delta(\mathcal{A})$ 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.

E.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 D.

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 D.

E.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 $s \in \{\text{ref, wrench, proprio, fuse}\}$ 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.

E.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:

$$h_{action} = \text{SiLU}(W^{(1)}_{action} h_{context} + b^{(1)}_{action}) \in \mathbb{R}^{128},$$

$$[\mu, \log \sigma] = W^{(2)}_{action} h_{action} + b^{(2)}_{action} \in \mathbb{R}^{14}$$

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.

E.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 $\beta^{(obj)}$.

E.2 Training Configuration

E.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).

E.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 $[\text{Softplus}(-10), \text{Softplus}(10)] \approx [4.5 \times 10^{-5}, 10.00]$)

E.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:

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

F   SE(2) Simulation Environment Settings

This appendix specifies the simulation environment configuration, task specifications, and evaluation protocol for the SE(2) planar manipulation experiments.

F.1   Simulation Environment

Workspace Configuration:

- Total space: 512×512 pixels
- Effective space: 501×501 pixels
- Walls: Located at $(5, 5) \sim (506, 506)$ with 2-pixel thickness

Physics and Control:

- Physics timestep: 0.002s (500 Hz simulation)
- Control frequency: 50 Hz (policy and controller)
- Robot platform: Dual end-effectors with direct wrench control

- Force-torque sensing: 3-axis (m_z, f_x, f_y) per end-effector at 50 Hz

Object Dataset:

- Total objects: 9 (3 joint types × 3 variants per type)
- Joint types: Fixed (rigid transport), Revolute (angular articulation), Prismatic (linear articulation)
- Material: Rigid body with Coulomb friction ($\mu = 0.6$–0.9)
- Contact model: Soft contact with 5000 N/m stiffness

F.2   Task Specifications

Task Objective: Manipulate an articulated object from a randomized initial configuration $({}^sT_o^{init}, q_{obj}^{init})$ to a fixed goal configuration $({}^sT_o^{goal}, q_{obj}^{goal})$ through coordinated bimanual control.

Object Configuration:

- Object frame pose: ${}^sT_o \in SE(2)$ (3 DoF: position (x_o, y_o), orientation θ_o)
- Joint state: $q_{obj} \in \mathbb{R}$ (1 DoF: angle for revolute, extension for prismatic)

Task Structure:

- Objects: 9 total, each with one fixed goal configuration
- Episode initialization: Object spawns at random configuration within safe workspace region
- Episode duration: 8 seconds (400 timesteps at 50 Hz)
- Visual feedback: Goal configuration marked with distinct pixel color in 512×512 workspace image

Joint Type Categories:

1. **Fixed Joint:** q_{obj} constant; manipulate ${}^sT_o$ to ${}^sT_o^{goal}$
2. **Revolute Joint:** Manipulate both ${}^sT_o$ and q_{obj} (rotation) to goal
3. **Prismatic Joint:** Manipulate both ${}^sT_o$ and q_{obj} (extension) to goal

Success Criteria (unified):

- Object position error: $\|p_o - p_o^{goal}\|_2 < 10$ pixels
- Object orientation error: $|\theta_o - \theta_o^{goal}| < 5$
- Joint error: $|q_{obj} - q_{obj}^{goal}| < 5$ (revolute) or < 5 pixels (prismatic)
- Grasp stability: Maintained throughout episode (no drift exceeding $d_{\max}$)

Termination Conditions:

- **Success:** All success criteria satisfied at episode end
- **Failure (grasp loss):** Grasp drift exceeds threshold: $\left\| \left[\log \left((T_{\text{grip},i}^{\text{init}})^{-1} T_{\text{grip},i} \right) \right]^\vee \right\|_2 > d_{\max}$ for any $i \in \{l, r\}$
- **Timeout:** Maximum episode duration (8 seconds / 400 timesteps) reached without success

F.3   Training Randomization

To improve policy robustness, we apply the following randomization during training:

- Object spawn: Uniform random within safe workspace region (with wall margin)
- Object orientation: Uniform random $\theta_o \in [0, 2\pi)$
- Joint configuration: Uniform random within joint limits
- End-effector positions: Random grasp configurations on object

F.4 Evaluation Protocol

Test Configuration:

- Objects: 9 (3 per joint type)
- Goal configurations: 1 fixed goal per object
- Trials: 100 rollouts per object with randomized initial configurations

Primary Metrics:

1. **Success Rate (%):** Binary task completion metric

$$\text{Success Rate} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}[\text{task}_i \text{ succeeded}] \tag{77}$$

   Confidence intervals computed via bootstrap (10,000 resamples).

2. **Constraint Violation (pixels/s):** Time-averaged kinematic constraint violation

$$\text{CViol} = \frac{1}{T} \sum_{t=1}^{T} \|{}^s\mathcal{V}_l(t) - {}^s\mathcal{V}_r(t) - \mathcal{S}\dot{q}_{obj}(t)\|_2 \tag{78}$$

   where $\mathcal{S}$ is the spatial screw axis and $T = 400$ timesteps.

3. **Internal Force (N):** Time-averaged non-productive contact forces

$$F_{\text{int}} = \frac{1}{T} \sum_{t=1}^{T} \sum_{i \in \{l,r\}} \|\mathcal{F}_{i,\perp}(t)\|_2 \tag{79}$$

   where $\mathcal{F}_{i,\perp}$ is the wrench component orthogonal to screw axis.

Secondary Metrics:

- **Peak Contact Force (N):** $F_{\text{peak}} = \max_{t,i} \|\mathcal{F}_i(t)\|_2$
- **Tracking RMSE (pixels):** End-effector tracking error

$$\text{RMSE} = \sqrt{\frac{1}{2T} \sum_{t=1}^{T} \sum_{i \in \{l,r\}} d_{SE(2)}^2(T_{si}^{actual}(t), T_{si}^{des}(t))} \tag{80}$$

- **Motion Smoothness (pixels/s³):** Jerk magnitude

Statistical Testing:

- Method: Welch's t-test ($p < 0.05$) with Bonferroni correction for multiple comparisons
- Effect size (Cohen's d): negligible ($|d| < 0.2$), small ($0.2 \le |d| < 0.5$), medium ($0.5 \le |d| < 0.8$), large ($|d| \ge 0.8$)

G   Pure Imitation Learning Baseline (OWIL)

G.1   Overview

extbfOWIL (Object-Wrench conditioned Imitation Learning) is a pure imitation learning baseline designed to evaluate the necessity of SWIVL's explicit kinematic decomposition and constraint-based action space. OWIL learns to satisfy kinematic constraints **implicitly** through behavior cloning on expert demonstrations, without explicit bulk–internal motion decomposition or kinematic-constrained action parameterization.

extbfPurpose: Answer Q1 by comparing SWIVL's physics-aware approach against state-of-the-art imitation learning that receives the same object and wrench information but relies on implicit constraint learning.

G.2 Architecture (SE(2) Version)

G.2.1 Observation Space

OWIL receives similar object and wrench information as SWIVL's low-level policy, but **excludes reference motion inputs**:

1. **Object Constraints** (6-dim):
   - Body-frame screw axes: $\mathcal{B}_l, \mathcal{B}_r \in \mathbb{R}^3$ (6-dim)

2. **Wrench Feedback** (6-dim):
   - Body wrenches: $\mathcal{F}_l, \mathcal{F}_r \in \mathbb{R}^3$ ((m_z, f_x, f_y) × 2 arms = 6-dim)
   - Filtered with exponential smoothing (α=0.3) to reduce sensor noise

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

Total observation dimension: $o_t \in \mathbb{R}^{24}$

Key Difference from SWIVL: No reference twist inputs $\{\mathcal{V}_l^{\text{ref}}, \mathcal{V}_r^{\text{ref}}\}$ (6-dim excluded). OWIL learns manipulation directly from demonstrations without explicit reference motion guidance from the vector field.

G.2.2 Action Space

OWIL uses a **direct arm twist action space** in SE(2) without kinematic constraint parameterization:

$$a_t = ([\mathcal{V}_l], [\mathcal{V}_r]) \in \mathfrak{se}(2) \times \mathfrak{se}(2)$$

Policy output: $(\mathcal{V}_l, \mathcal{V}_r) \in \mathbb{R}^3 \times \mathbb{R}^3$ (6-dim)

- Left arm twist: $(\omega_{z,l}, v_{x,l}, v_{y,l})$ (3-dim)
- Right arm twist: $(\omega_{z,r}, v_{x,r}, v_{y,r})$ (3-dim)

Constraint Handling: The kinematic constraint $\mathcal{V}_l - \mathcal{V}_r = \mathcal{S}\dot{q}_{obj}$ is **not enforced structurally**. OWIL must learn to satisfy constraints implicitly through demonstrations.

G.2.3 Network Architecture

Encoders:

- **Object encoder (FiLM conditioning):** [64, 64] $\rightarrow$ 128-dim $\rightarrow$ generates $(\gamma, \beta) \in \mathbb{R}^{64} \times \mathbb{R}^{64}$
- **Wrench encoder:** [64, 64] $\rightarrow$ 64-dim
- **Proprioception encoder:** [128, 128] $\rightarrow$ 128-dim

Policy Head:

- Fusion layer: Concatenate [wrench(64) + proprio(128)] $\rightarrow$ 192-dim
- FiLM modulation: $\text{FiLM}(x) = \gamma \odot x + \beta$
- Residual MLP: [256, 256] $\rightarrow$ 6-dim twist outputs (3-dim per arm)

G.3 Training

G.3.1 Data Collection

- **Source:** Human teleoperation demonstrations
- **Tasks:** All 9 objects (3 fixed, 3 revolute, 3 prismatic) with diverse initial configurations

- **Demonstrations:** 5,000 expert trajectories collected across all objects
- **Augmentation:** Random spawn within safe workspace region, uniform orientation $\theta_o \in [0, 2\pi)$, random joint configurations within limits

G.3.2 Training Protocol

- **Algorithm:** Behavior cloning with MSE loss
- **Optimizer:** Adam (lr=1e-4, β=(0.9, 0.999))
- **Batch size:** 256 trajectories
- **Training steps:** 100k gradient updates
- **Loss function:**

$$\mathcal{L}_{\text{BC}} = \mathbb{E}_{(o,a)\sim\mathcal{D}} \left[\|\pi_\theta(o) - a^{\text{expert}}\|_2^2 \right]$$

G.4 Key Differences from SWIVL

Component	OWIL	SWIVL
Learning Paradigm	Behavior cloning (offline)	Reinforcement learning (online)
Reference Motion	None (direct from demonstrations)	Stable imitation vector field
Action Space	Direct arm twists $(\mathcal{V}_l, \mathcal{V}_r)$	Impedance variables $(d_{l,\|}, d_{r,\|}, d_{l,\perp}, d_{r,\perp}, k_{p_l}, k_{p_r}, \alpha)$
Constraint Satisfaction	Implicit learning from data	Structural guarantee via projection operators
Observation Dimension	24-dim (no reference)	30-dim (reference included)
Action Dimension	6-dim (unconstrained twists)	7-dim (impedance modulation)

G.5 Expected Limitations

1. **Constraint Violations:** Without structural constraint enforcement, OWIL may violate kinematic constraints, especially when deviating from training distribution.

2. **Internal Forces:** Implicit coordination learning may lead to higher internal forces compared to SWIVL's explicit force decomposition and compliance.

3. **Generalization:** Pure imitation may overfit to demonstration characteristics, limiting adaptation to novel objects or task variations.

4. **Data Efficiency:** Requires large amounts of expert data to implicitly capture constraint satisfaction patterns.

5. **Robustness:** No corrective feedback mechanism (no vector field stability) when execution deviates from demonstrated trajectories.

G.6 Evaluation Protocol

OWIL is evaluated using the same metrics as SWIVL:

Primary Metrics:

1. **Success Rate (%):** Task completion within error thresholds (position < 10 pixels, orientation < 5, joint < 5 or < 5 pixels)

2. **Constraint Violation (pixels/s):** $\text{CViol} = \frac{1}{T} \sum_t \|\mathcal{V}_l - \mathcal{V}_r - \mathcal{S}\dot{q}_{obj}\|_2$

3. **Internal Force (N):** $F_{int} = \frac{1}{T} \sum_{t,i} \|\mathcal{F}_{i,\perp}\|_2$ where $\mathcal{F}_{i,\perp}$ is the wrench component orthogonal to screw axis

Evaluation: 9 objects $\times$ 100 trials per object = 900 rollouts per method.

extbfHypothesis: OWIL's implicit constraint learning will show higher constraint violations and internal forces compared to SWIVL's explicit physics-aware approach, especially on novel objects and task variations not well-represented in demonstrations.