Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

Tune to Learn: How Controller Gains Shape Robot Policy Learning

Antonia Bronars*
MIT
*Equal Contribution

Younghyo Park*
MIT
*Equal Contribution

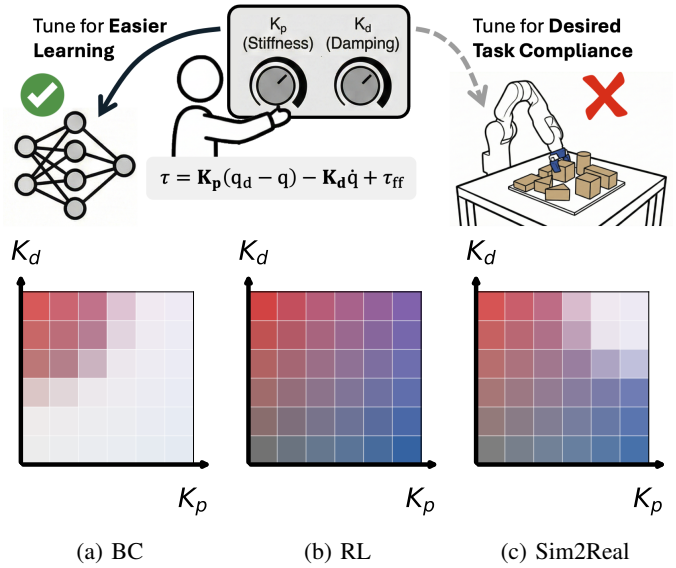
Pulkit Agrawal
MIT
Improbable AI Lab

***Abstract*—Position controllers have become the dominant interface for executing learned manipulation policies. Yet a critical design decision remains understudied: how should we choose controller gains for policy learning? The conventional wisdom is to select gains based on desired task compliance or stiffness. However, this logic breaks down when controllers are paired with state-conditioned policies: effective stiffness emerges from the interplay between learned reactions and control dynamics, not from gains alone. We argue that gain selection should instead be guided by learnability: how amenable different gain settings are to the learning algorithm in use. In this work, we systematically investigate how position controller gains affect three core components of modern robot learning pipelines: behavior cloning, reinforcement learning from scratch, and sim-to-real transfer. Through extensive experiments across multiple tasks and robot embodiments, we find that: (1) behavior cloning benefits from compliant and overdamped gain regimes, (2) reinforcement learning can succeed across all gain regimes given compatible hyperparameter tuning, and (3) sim-to-real transfer is harmed by stiff and overdamped gain regimes. These findings reveal that optimal gain selection depends not on the desired task behavior, but on the learning paradigm employed. Project website: https://younghyopark.me/tune-to-learn**

I. INTRODUCTION

Position controllers are rapidly becoming the de facto choice for low-level control in robot learning. Their wide hardware support and intuitive nature have made them the dominant interface for executing learned manipulation policies. Yet while classical control theory provides clear guidance on selecting gains to achieve desired tracking bandwidth, disturbance rejection, or impedance characteristics, no analogous principles exist for the learning setting. An important design decision remains overlooked: how should we choose controller gains when *learning* data-driven manipulation policies?

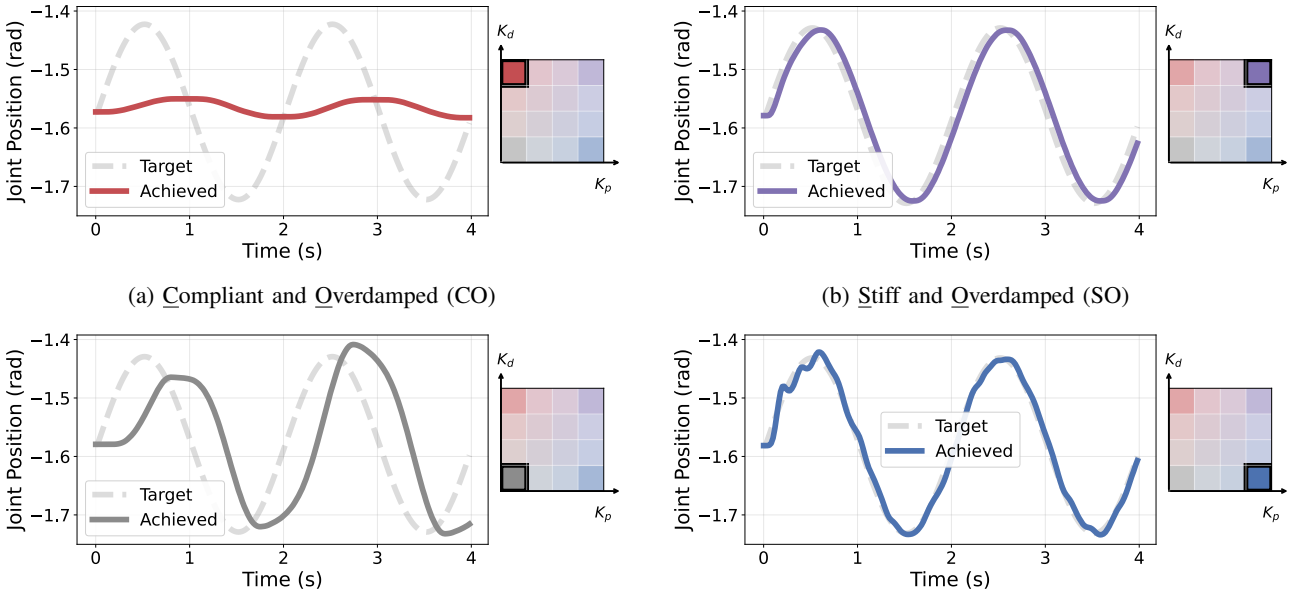
The standard approach treats gain selection as a problem of achieving desired task behavior—contact-rich manipulation calls for low stiffness and high damping to better comply with unexpected contacts, while precision tasks call for high stiffness and low damping to accurately track position commands. But this framing conflates two distinct roles that position controllers play. When tracking open-loop trajectories, the controller *is* the *behavior*—gains directly determine how the robot responds. When paired with a learned policy, however, the controller becomes an *interface* between the policy and the physical world. The policy learns through this interface during training and acts through this interface at deployment. Viewed this way, gains function less as behavioral parameters and more as an *inductive bias*—an implicit prior over the space of closed-loop behaviors that shapes what the policy can easily express and learn.



(a) BC (b) RL (c) Sim2Real

Fig. 1: **Different robot learning paradigms prefer different controller gain interfaces.** Colored regions indicate gain regimes where each paradigm succeeds. Contrary to conventional wisdom of tuning gains for desired task compliance, optimal gains depend on the learning paradigm. Based on our experimental findings, heatmaps illustrate representative gain preferences for (a) behavior cloning, which favors compliant, overdamped gains, (b) reinforcement learning, which adapts to nearly any setting, and (c) sim-to-real transfer, which is degraded by stiff and overdamped gains.

This distinction matters because learned policies are reactive: they observe the current state and output corrective commands. A policy can achieve stiff or compliant task-level behavior regardless of the underlying joint-level gains, simply by modulating the magnitude and timing of its outputs. The gains, therefore, do not determine the set of achievable closed-loop behaviors. We hypothesize that the gains instead shape the learning problem: how easy it is to fit action labels and how errors compound during closed-loop execution, which training configurations yield successful RL policies, and whether modeling discrepancies amplify into instability during sim-to-real transfer.



(a) Compliant and Overdamped (CO)

(b) Stiff and Overdamped (SO)

(c) Compliant and Underdamped (CU)

(d) Stiff and Underdamped (SU)

Fig. 2: **Controller gains induce diverse action–response dynamics.** We evaluate a broad range of representative gain configurations and their resulting dynamic responses to assess their impact on learnability.

Once we recognize controller gains as learning interface parameters rather than behavioral parameters, the design question becomes: which interface properties facilitate learning? And critically, do different learning paradigms prefer different interfaces, serving as a conducive *inductive bias*? We investigate these questions systematically across three paradigms of modern robot learning and present the following findings:

1) Behavior cloning performs best with *compliant* and *overdamped* gains. Across multiple manipulation tasks with controlled datasets that isolate the effect of gains, we show this regime yields higher closed-loop policy success rates without penalizing teleoperation efficiency. (Sec. IV-A and V-A)
2) Reinforcement learning (RL) from scratch is agnostic to gain setting, as long as the remaining hyperparameters are tuned to be compatible with the given gain setting. We verify this by obtaining equivalently successful RL policies for all gain regimes across multiple manipulation and locomotion tasks. (Sec. IV-B and V-B)
3) When transferring policies from simulation to the real-world, *stiff* and *overdamped* controllers exacerbate the motor-level sim-to-real gap. (Sec. IV-C and V-C)

Our findings converge on a unified picture of how controller gains shape learning, providing both conceptual clarity and practical guidance for this widely used yet underexplored design decision.

II. Related Works

A. Position and Impedance Control

Position and impedance control have long been foundational for robot manipulation. Consider a robot manipulator with joint positions $\mathbf{q} \in \mathbb{R}^n$ governed by the dynamics:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{ext}} \tag{1}$$

where $\mathbf{M}(\mathbf{q})$ is the inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ captures Coriolis and centrifugal effects, $\mathbf{g}(\mathbf{q})$ is the gravity vector, $\boldsymbol{\tau}$ is the control torque, and $\boldsymbol{\tau}_{\text{ext}}$ represents environmental torques.

Takegaki and Arimoto [1] established the global asymptotic stability of PD control with gravity compensation:

$$\boldsymbol{\tau} = \mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) + \mathbf{K}_d(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) \tag{2}$$

where $\mathbf{K}_p, \mathbf{K}_d \in \mathbb{R}^{n \times n}$ are gain matrices representing joint stiffness and damping, respectively, and $\mathbf{q}_d, \dot{\mathbf{q}}_d$ are the desired joint positions and velocities. This control law can be interpreted as *joint-space impedance control*: in the absence of external torques, the closed-loop system behaves as a virtual spring-damper attached to the desired configuration, with the impedance relationship:

$$\boldsymbol{\tau}_{\text{ext}} = \mathbf{K}_p(\mathbf{q} - \mathbf{q}_d) + \mathbf{K}_d(\dot{\mathbf{q}} - \dot{\mathbf{q}}_d). \tag{3}$$

This formulation is prevalent in modern robot learning, where policies typically output joint position targets $\mathbf{q}_d$ that are tracked by a low-level PD controller. Kelly [2] provided a comprehensive review analyzing equilibrium uniqueness and stability robustness against parametric uncertainties. Despite the theoretical foundations, gain selection remains largely heuristic in practice.

B. Low-Level Control in Robot Learning

Action Spaces. Position-controlled action spaces (whether commanding joint positions or end-effector poses) convert policy outputs to motor torques via feedback control laws,

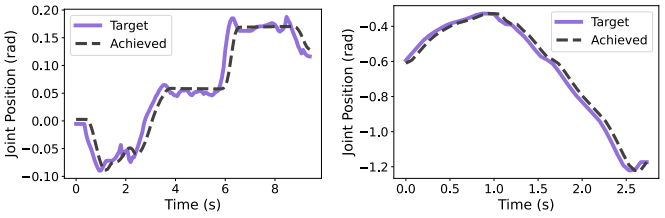
making controller gains an implicit component of action space design. Aljalbout *et al.* [3] find that action spaces based on control abstractions (e.g., PD-controlled positions) generally outperform torque control, though they do not vary gains within each paradigm. Kim *et al.* [4] argue that torque control's inherent compliance mitigates the sim-to-real gap. Eßer *et al.* [5] frame action space selection as an *inductive bias* for locomotion—a perspective we extend to gain selection within position-controlled manipulation. Our study complements these works by isolating the effect of PD gains while holding the action representation fixed.

Learning Task-level Impedance Policies. Several works train policies that exhibit compliant manipulation behavior, either by learning variable stiffness profiles from demonstrations [6, 7] or by conditioning on user-specified task-space stiffness [8]. Notably, Margolis *et al.* [8] observe that a policy's compliance is dictated by its training incentives, not the underlying PD gains, i.e., a policy can learn soft or stiff interactions regardless of the low-level controller. This distinction motivates our study: rather than learning compliance as a task-level behavior, we investigate how *fixed* gain settings shape the learning process itself. Arachchige *et al.* [9] also vary gains of their underlying position controllers, but focus on speeding up execution of pretrained policies rather than understanding how gains affect learning.

Sim-to-Real Transfer and Controller Fidelity. Domain randomization [10, 11] has become standard for bridging the sim-to-real gap, with dynamics randomization typically including controller-related parameters such as PD gains, motor strengths, and joint damping [12]. Muratore *et al.* [13] provide a comprehensive review of sim-to-real transfer via randomized simulations, noting that contact and friction models (which interact strongly with controller gains) remain among the most challenging aspects to transfer. Control frequency has also been shown to affect transfer fidelity: Gangapurwala*et al.* [14] show that low-frequency policies are less sensitive to actuation dynamics, enabling successful sim-to-real transfer without dynamics randomization. Despite this, systematic study of how the nominal gain settings (around which randomization occurs) affect sim-to-real transfer is lacking. Understanding which gain regimes transfer more robustly can inform both the choice of nominal gains and the design of more targeted randomization ranges, rather than treating all gain configurations as equally viable starting points.

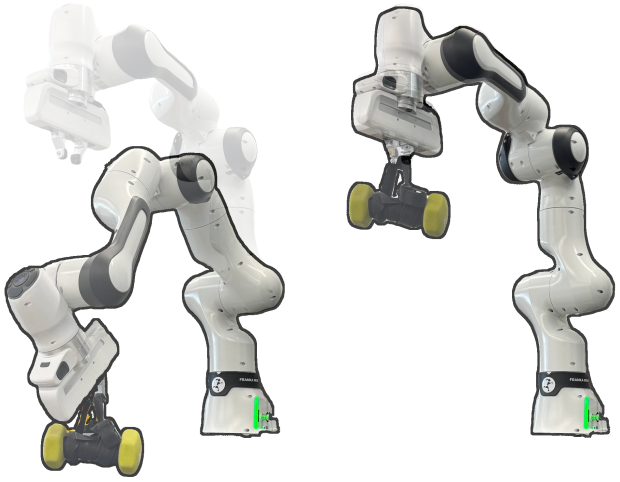
C. Gain Settings in Large-Scale Robot Datasets

While controller gains fundamentally shape the learning interface, their configuration in existing large-scale datasets remains largely undocumented. To understand current practices, we analyzed DROID [15] and several datasets within the Open X-Embodiment collection [16] by examining the relationship between commanded and achieved joint positions. Although exact gain values are rarely reported, tracking behavior reveals controller characteristics. As shown in Fig. 3, achieved positions closely track commands with minimal lag and overshoot,



(a) DROID (b) RT-X NYU Franka Play

Fig. 3: Tracking response curves from existing robot datasets reveal tight command-following behavior, suggesting stiff controller gains are prevalent in existing data collection pipelines.



(a) Compliance w/ stiff gain (b) Stiffness w/ compliant gain

Fig. 4: **Task-level impedance can be decoupled from low-level controller gains with learned policies.** A learned policy can achieve (a) *compliant* behavior despite stiff low-level gains, and (b) *stiff* behavior despite compliant gains.

characteristic of stiff controllers. This pattern was prevalent across datasets, suggesting stiff gains have become an implicit default in data collection.

III. Decoupling Gains from Task Compliance

In this section, we validate a central claim: *A policy's task-level compliance is predominantly determined by its learned reactions, rather than its underlying gains.* In this section, we validate this claim empirically through two intentionally counterintuitive pairings:

Stiff behavior with compliant gains. We train a reinforcement learning policy to maintain a fixed pose under external disturbances. Although the low-level controller operates with compliant (low-gain) impedance, we induce stiff task-level behavior by randomly applying force disturbances during training and rewarding the policy for remaining close to a target pose. Specifically, we use a sharp distance-based reward, i.e.,

$$r(\mathbf{q}) = 1 - \tanh(\|\mathbf{q} - \mathbf{g}\|^2/\lambda) \tag{4}$$

where a small λ strongly penalizes deviations from the goal. This encourages the policy to actively counteract disturbances and maintain its pose, resulting in stiff task-level responses despite compliant low-level control, as shown in Fig. 4b.

Compliant behavior with stiff gains. To elicit compliant task-level responses despite a stiff (high-gain) low-level controller, we soften the goal-tracking objective and discourage aggressive controller corrections:

$$r(\mathbf{q}) = 1 - \tanh(\|\mathbf{q} - \mathbf{g}\|^2 / \lambda_{\text{large}}) - \alpha \|\Delta a_t\|^2. \quad (5)$$

A large λ reduces the incentive to tightly regulate the goal pose, while the Δa penalty suppresses high-frequency corrective behavior, encouraging the policy to yield smoothly under disturbances even when executed with stiff gains (Fig. 4a). We provide quantitative support for this in Appendix A.

IV. Experiments

In this section, we detail the experimental procedures we used to study the effect of low-level position controllers on learning dynamics for behavior cloning (Sec. IV-A), online reinforcement learning (Sec. IV-B), and zero-shot sim-to-real transferability (Sec. IV-C). Candidate gain setpoints used for all analysis are represented as a grid of $\mathbf{K}_p$ and $\mathbf{K}_d$ as shown in Fig. 2.

A. Behavior Cloning

We investigate how controller gains affect behavior cloning performance and data collection experience through controlled dataset generation and a user study.

Gain-Dependent Demonstration Dataset. Behavior cloning distills state-action pairs $\mathcal{D}(s, a)$ into a policy $\pi(a|s)$. With position targets as actions, the controller gains $\mathbf{K} = (\mathbf{K}_p, \mathbf{K}_d)$ implicitly shape learning by altering the state transition dynamics $p(s'|s, a)$. Isolating this effect requires datasets $\mathcal{D}(s, a; \mathbf{K})$ that share a common state distribution while exhibiting gain-induced action distributions $p(a; \mathbf{K})$.

Naively collecting separate demonstrations per gain setting conflates state and action variation: differing closed-loop dynamics and collection stochasticity (initial conditions, environmental randomness, demonstrator variability) cause both distributions to shift, obscuring the effect of gains on learning.

Torque-to-Position Retargeting. We instead achieve nearly identical state trajectories across all gain settings while varying only the position target actions through Torque-to-Position Retargeting (TPR), a two-stage dataset generation procedure. First, we generate demonstration trajectories for each task at high frequency (500Hz) using *torque* commands as the gain-agnostic action representation. We then apply a position-command-retargeting method to convert these torque trajectories into position targets for arbitrary $(\mathbf{K}_p, \mathbf{K}_d)$ settings:

$$\mathbf{q}_{\text{des}}(t) = \mathbf{q}(t) + \mathbf{K}_p^{-1} \left(\boldsymbol{\tau}(t) + \mathbf{K}_d \dot{\mathbf{q}}(t) \right), \quad (6)$$

where $\tau(t), q(t)$ and $\dot{q}(t)$ are the torque command, joint position, and joint velocity from the original 500Hz torque control demonstration, respectively. Finally, for each gain configuration, we replay the retargeted position commands at the desired policy frequency (50Hz) using zeroth-order holding and save only the successful rollouts, ensuring that our datasets capture the same task outcomes across different controller settings while maintaining the distinct action distributions induced by each gain profile. We conduct this entire process in simulation to ensure controlled experimental conditions. We quantitatively validate TPR fidelity: retargeted trajectories maintain $\geq 90\%$ success rate and joint-position MSE $< 10^{-3}$ across gain configurations up to $25\times$ decimation (20 Hz); at higher decimation, success degrades slightly for contact-rich tasks where TPR's trajectory-matching assumption is less robust (Appendix B5).

Extension to Task-Space Position Control. While the above formulation addresses joint-space position control, many manipulation systems instead use operational space control (OSC) [17] with SE(3) end-effector pose targets. OSC computes joint torques through a task-space impedance law:

$$\boldsymbol{\tau} = \mathbf{J}^\top \mathbf{M}_x \left(\mathbf{K}_p \tilde{\mathbf{x}} - \mathbf{K}_d \dot{\mathbf{x}} \right) + \boldsymbol{\tau}_{\text{null}}, \quad (7)$$

where $\tilde{\mathbf{x}}$ is the pose error (position and orientation), $\dot{\mathbf{x}}$ is the task-space velocity, and $\mathbf{M}_x$ is the task-space inertia matrix. The gains $\mathbf{K}_p, \mathbf{K}_d \in \mathbb{R}^{6\times 6}$ now operate in Cartesian space, separately controlling translational and rotational compliance.

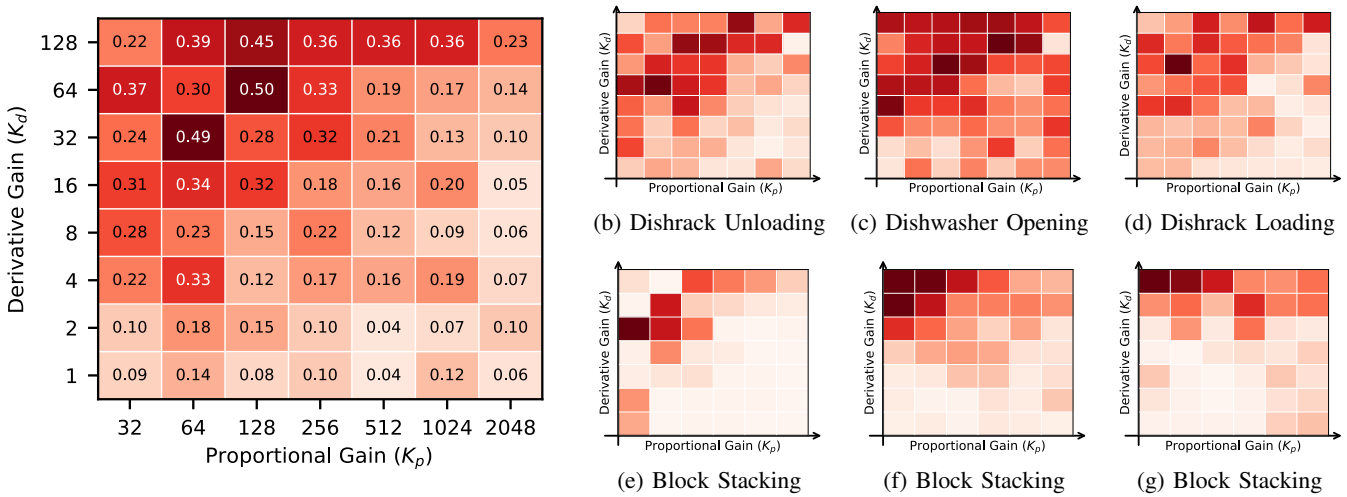
TPR extends naturally to this setting. We collect demonstrations using torque control and record the task-space wrench $\mathbf{F}(t) = \mathbf{M}_x(\mathbf{K}_p \tilde{\mathbf{x}} - \mathbf{K}_d \dot{\mathbf{x}})$ along with the end-effector pose $\mathbf{x}(t)$ and velocity $\dot{\mathbf{x}}(t)$. Retargeting to a new gain configuration $(\mathbf{K}'_p, \mathbf{K}'_d)$ then yields:

$$\mathbf{x}_{\text{des}}(t) = \mathbf{x}(t) + \mathbf{K}'^{-1}_p \left(\mathbf{F}(t) + \mathbf{K}'_d \dot{\mathbf{x}}(t) \right), \quad (8)$$

where the orientation component is handled by converting the resulting axis-angle error to a quaternion displacement.

Training Configurations. We then train BC policies for each gain configuration $\mathbf{K} \in \{\mathbf{K}_1, \cdots, \mathbf{K}_n\}$, using gain-dependent demonstration datasets $\mathcal{D}(s, a(\mathbf{K}))$. Our nominal configuration uses a VAE generative model with an MLP network, observation history length 10, and action chunk size 10. We use privileged simulation states (i.e. object poses) as inputs, and absolute joint-space actions as outputs. To verify that our findings are not artifacts of this particular setup, we ablate across network architectures (MLP vs. Transformer), policy model classes (regression, VAE, and diffusion [18]), temporal structure (observation history length and action chunk size), input modalities (privileged simulation states vs. robot state with dual-camera RGB from global and wrist-mounted views), and output representations (joint-space vs. task-space actions). We verify that gain preferences are consistent across all variations; full ablation results are reported in Appendix B.

Gain-Dependent Teleoperation User Study. To complement our analysis on offline policy training, we conducted a user study examining how controller gains affect human teleoperation performance. We designed a contact-rich, non-prehensile box manipulation task: operators teleoperate a Franka Research 3 robot with a 6-DoF SpaceMouse to push a box from a



(a) Bimanual Handover

(b) Dishrack Unloading (c) Dishwasher Opening (d) Dishrack Loading

(e) Block Stacking w/ Task Space Control (f) Block Stacking /w GravComp (g) Block Stacking /wo GravComp

Fig. 5: **Behavior cloning prefers *compliant* and *overdamped* controller gains.** Closed-loop rollout success rates across a grid of proportional ($\mathbf{K}_p$) and derivative ($\mathbf{K}_d$) gains for diverse manipulation tasks and robot embodiments. Each heatmap reports success averaged over evaluation rollouts. Across tasks, higher success rate (darker red) consistently concentrates in the compliant, overdamped regime (upper-left), while stiff or weakly damped controllers yield degraded performance.

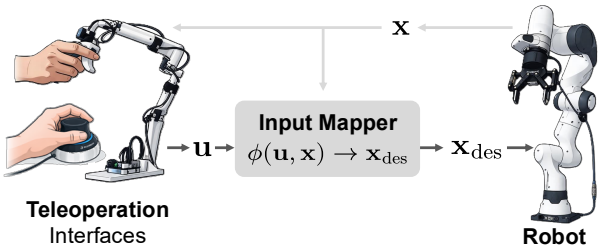


Fig. 6: Any teleoperation system requires a mapping ϕ from user inputs $\mathbf{u}$ to desired robot targets $\mathbf{x}_{\text{des}}$, which substantially shapes how the robot is perceived under different controller gains.

randomized initial pose to a fixed goal pose. We chose this task because it requires both precision and sustained contact, yet remains achievable even under unintuitive gain configurations, allowing us to compare gain settings without floor effects.

A critical consideration is that the mapping from user input to commanded position target, $\phi(\mathbf{u}, \mathbf{x}) \rightarrow \mathbf{x}_{\text{des}}$ (Fig. 6), can substantially modulate how the robot feels to the teleoperator. In our teleoperation setup, the mapping takes the form $\mathbf{x}_{\text{des}}(t) = \alpha\mathbf{u}(t) + \beta\mathbf{x}(t) + (1-\beta)\mathbf{x}_{\text{des}}(t-1)$, where $\alpha \in \mathbb{R}_0^+$ is a scaling factor and $\beta \in \{0, 1\}$ selects between the current robot pose or the previous target as the integration base. To ensure a fair comparison, each user study participant adjusts α and β during a practice period before evaluation for each gain setting, yielding the gain-specific optimum

$$\phi^\star(\mathbf{K}) = \arg\max_{\phi} \mathcal{Q}(\phi; \mathbf{K}), \tag{9}$$

where $\mathcal{Q}$ denotes the operator's perceived control quality. This lets operators compare each gain configuration at its best achievable experience.

We asked 12 users to perform non-prehensile box pushing task over 1-hour sessions, collecting 1,297 total trials. Gain configurations were randomly sampled and blindly presented for each trial to control for learning effects across the session. Trials fail if the robot faults (position, velocity, or torque limit violation) or the operator pushes the box out of the workspace. For each trial, we recorded task success, completion time, and a subjective 1–5 control quality rating. The results are presented in Result V-A-II.

B. Online Reinforcement Learning

In this section, we study how controller gains affect online RL, where gains shape both the transition dynamics and exploration during training.

Gain-Dependent Environment Shaping. A key challenge in isolating the effect of controller gains on online RL is that performance can be highly sensitive to environment design and algorithm hyperparameters. These choices collectively determine the learning regime, a dependence we refer to as *environment shaping* [19]. To avoid conflating gain effects with suboptimal training configurations, we re-tune key hyperparameters for each gain setting using computational hyperparameter optimization [20][1][2]:

$$h^\star(\mathbf{K}) = \arg\max_{h} \; J\big(\pi^\star(h; \mathbf{K})\big), \tag{10}$$

where $\pi^\star(h; \mathbf{K})$ denotes the converged policy under gains $\mathbf{K}$ and hyperparameters h. This ensures each gain configuration is evaluated at its best achievable performance, allowing us to determine whether RL can discover successful behaviors

[1]Specifically, we optimize the action space design parameters $h := (\alpha, \beta, \gamma)$ in $\mathbf{x}_{\text{des}}(t) = \alpha\mathbf{u}(t) + \gamma\beta\mathbf{x}(t) + \gamma(1-\beta)\mathbf{x}_{\text{des}}(t-1)$, where $\mathbf{u}(t)$ is the policy output, $\alpha \in \mathbb{R}_0^+$ scales the policy output, γ selects between absolute (γ=0) and relative (γ=1) actions, and β determines whether relative actions are integrated on current state (β=1) or the previous target (β=0).

[2]Policies are trained using the SKRL implementation [21] of PPO [22] on tasks modified from IsaacLab [23].

regardless of gain settings. Findings on solution existence are presented in Result V-B-I.

Hyperparameter Optimization Landscape. Beyond solution existence, we also investigate whether certain gain regimes make hyperparameter optimization easier. We consider gain regimes advantageous if they yield large, continuous regions of successful hyperparameters that are easily discoverable via optimization. To investigate how gain settings modulate the shape of this successful region, we record success rates across the hyperparameter landscape during 50 trials per gain setting, continuing all trials to completion even after finding a working configuration. Analysis of the optimization landscape is presented in Result V-B-II.

Sample Efficiency. In additon to the hyperparameter landscape, we also investigate whether certain gain regimes yield more efficient or stable learning once a successful hyperparameter configuration has been identified. We consider a gain regime favorable if it enables policies to achieve high reward quickly and with low variance across random seeds. To isolate this effect, we run 5 random seeds for each hyperparameter combination that yielded >95% success during hyperparameter optimization, and compare the mean and standard deviation of training reward over the course of training, aggregated across all successful configurations. Analysis of sample efficiency and training stability under different gain regimes is presented in Result V-B-III.

C. Sim-to-Real

Finally, we examine whether certain gain settings transfer more reliably from simulation to real hardware. We study reaching tasks with a Franka Research 3 robot to directly isolate the motor-level sim-to-real gap.

System Identification. To ensure a fair comparison across gain settings, we first perform gain-specific system identification. For each gain configuration $\mathbf{K} \in \{\mathbf{K}_1, \cdots, \mathbf{K}_n\}$, we excite the real-world robot with sinusoidal position targets $\mathbf{q}^{\text{des}}(t) = A\sin(\omega t)$ and optimize simulation parameters ψ to match the resulting state trajectories, i.e.,

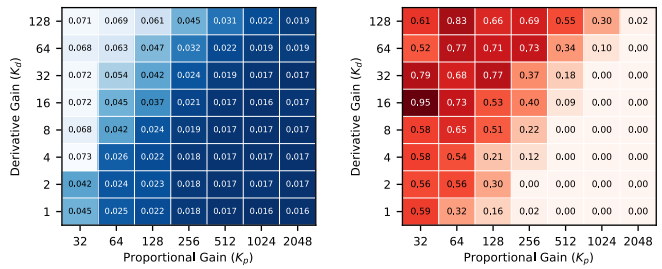
$$\psi^\star(\mathbf{K}) = \arg\min_{\psi} \sum_{t=0}^{T} \|\mathbf{x}(t;\mathbf{K}) - \bar{\mathbf{x}}(t;\psi)\|^2 \qquad (11)$$

where $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}})$ denotes the real robot state and $\bar{\mathbf{x}}(\cdot;\psi)$ its simulated counterpart with simulation parameters ψ.

This yields gain-dependent simulation environments that faithfully reproduce the closed-loop dynamics of each controller configuration as best as it can. Analysis of how gain settings affect system identification quality is presented in Result V-C-I.

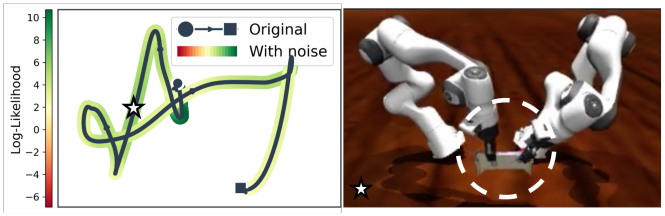
Gain-Dependent Sim-to-Real Transfer. For each gain setting, we train RL policies in the corresponding calibrated simulation environment. We discover succcessful and transferable solutions by adapting Eq. 10 as:

$$h^\star(\mathbf{K}) = \arg\max_{h} \; \tilde{J}\big(\pi^\star(h;\mathbf{K})\big), \qquad (12)$$

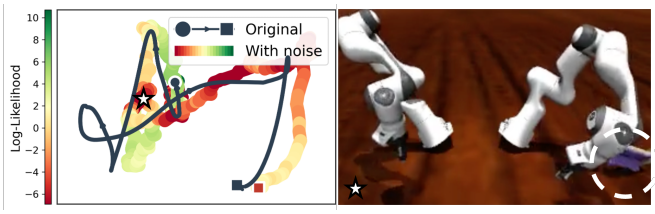


(a) Validation Loss

(b) Open-loop Success Rate with Action Noise



(c) Noised Open-loop Rollout for Compliant and Overdamped Gains



(d) Noised Open-loop Rollout for Stiff and Underdamped Gains

Fig. 7: **Compliant controllers attenuate action errors.** (a) Validation MSE loss during training: compliant gains yield higher loss, while stiff gains achieve lower loss. (b) Open-loop success rate under action noise: compliant gains maintain high success while stiff gains completely fail. (c) Compliant gains keep the perturbed trajectory close to the original, while (d) stiff gains cause large deviations that lead to task failure.

where $\tilde{J}$ augments the original objective with a penalty for violating real-world robot limits. We deploy the policies directly on the real robot without further fine-tuning; this zero-shot transfer protocol isolates the effect of gains on transferability. We also evaluate an ablation with domain randomization, where simulation parameters are perturbed within 10% of their system-identified values during training. We evaluate sim-to-real performance via *trajectory error*, i.e. the mean squared error (MSE) between real and simulated state trajectories when initialized from matched configurations:

$$\mathcal{E} = \underbrace{\|\mathbf{q}_{\text{sim}} - \mathbf{q}_{\text{real}}\|^2}_{\text{position error}} + \underbrace{\|\dot{\mathbf{q}}_{\text{sim}} - \dot{\mathbf{q}}_{\text{real}}\|^2}_{\text{velocity error}} \qquad (13)$$

For each gain setting, we report the average trajectory error across 30 real-world rollouts. Results on how gain settings affect zero-shot transfer performance are presented in Result V-C-II.

V. Results

A. Behavior Cloning

Gain settings influence behavior cloning performance in two ways: (1) through the controller's response to action prediction errors during closed-loop execution, and (2) through the controller's effect on the human demonstrator during teleoperated data collection. We study each factor in isolation.

> **Result V-A-I** (Effect on Learning): Under state-matched demonstrations (via TPR), behavior cloning performs best with *compliant* and *overdamped* gains (i.e., top left region of Fig. 2).

Using TPR to hold the state distribution constant across gain settings and vary only the action distribution, we consistently observe that *compliant* and *overdamped* gain setpoints yield significantly better closed-loop policy performance. Figure 5 illustrates this trend over a broad grid of controller gains and manipulation tasks. The same trend persists across diverse training configurations, including (a) state-based versus image-based policies, (b) action-chunked versus non–action-chunked policies, and (c) policies trained with or without state histories (Appendix B), as well as (d) task-space versus joint-space control (Fig. 5e), and (e) with or without gravity compensation (Fig. 5g). Additional results and visualizations are available on our project website.

To verify that the observed advantage of compliant, overdamped gains is statistically significant, we conduct formal hypothesis testing across all tasks. First, we fit a binomial logistic regression on $\log_2 \mathbf{K}_\mathrm{p}$ and $\log_2 \mathbf{K}_\mathrm{d}$ using $N{=}100$ rollouts per gain cell, confirming that lower $\mathbf{K}_\mathrm{p}$ and higher $\mathbf{K}_\mathrm{d}$ are significant predictors of success. We then apply Bonferroni-corrected one-sided Barnard's exact tests ($\alpha \approx 0.0083$, correcting for 6 tasks) under the null hypothesis

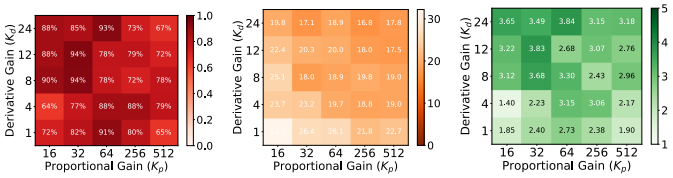
$$\mathcal{H}_0 \colon P(\text{success} \mid \mathcal{G}^{\text{CO}}) \leq P(\text{success} \mid \mathcal{G} \setminus \mathcal{G}^{\text{CO}}) \tag{14}$$

where $\mathcal{G}^{\text{CO}}$ denotes the compliant-overdamped gain region. $\mathcal{H}_0$ is rejected in all six tasks with $p \ll \alpha$ (Table II in Appendix B6).

Higher MSE Loss, Better Performance. Policies trained under compliant and overdamped gains exhibit higher training and validation MSE loss than those trained under stiff gains (Fig. 7a). Yet these same policies achieve higher closed-loop success rates. Lower imitation loss does not translate to better policy performance in our experiments.

Robustness to Action Noise. To further characterize this effect, we execute identical open-loop action sequences across all gain configurations while injecting random action noise at each timestep, with maximum noise magnitudes matched to the average validation loss observed during training (Fig. 7b). Under identical perturbations, compliant and overdamped gains maintain higher success rates than stiff gains.

Discussion. Together, these observations are consistent with the error-attenuation properties of compliant and overdamped controllers: low stiffness reduces the force produced by a given action error, while high damping dissipates perturbations faster, jointly limiting the resulting state deviation. This explains the apparent paradox — compliant gains produce action targets that are harder to fit (higher MSE), but the controller attenuates the resulting errors during execution, yielding better closed-loop performance.



(a) Success Rate (b) Completion Time (c) User Rating

Fig. 8: **Teleoperation performance under different gain regimes.** With optimized input mapping $\phi^\star(\mathbf{K})$ (Eq. 9), compliant and overdamped controllers (grid top-left) achieve similar or better success rates, user ratings, and shorter completion time to stiffer settings.

> **Result V-A-II** (Effect on Teleoperation): *Compliant* and *overdamped* gain settings yield comparable teleoperated data collection efficiency compared to other gain regimes, achieving similar yield and operator preference.

One might expect that the *compliant* and *overdamped* gains (favorable for policy learning) would hinder teleoperation, as sluggish robot response could frustrate operators and reduce collection efficiency. Our user study reveals that this is not necessarily the case (Fig. 8). When each gain configuration is evaluated with its own optimized input mapping $\phi^\star(\mathbf{K})$ (Eq. 9), compliant and overdamped settings achieve comparable or better success rates and receive high subjective ratings; appropriate tuning of the mapping function, particularly the scaling factor α, compensates for reduced responsiveness, allowing operators to command sufficiently responsive movements when needed. These results indicate that adopting compliant, overdamped gains for imitation learning does not impose a penalty on data collection. Full experimental details are in Appendix C.

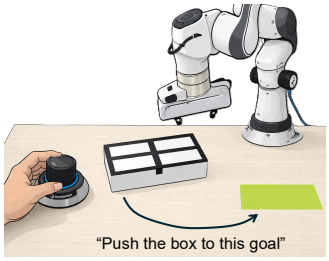
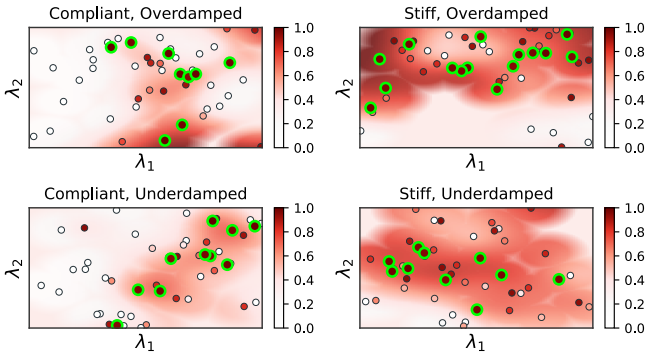


Fig. 9: Users were asked to perform nonprehensile pushing task under randomly sampled, blindly presented gain settings.

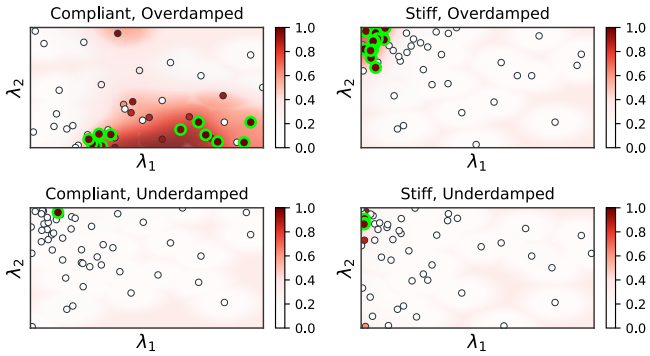
B. Online Reinforcement Learning

> **Result V-B-I** (RL Solution Existence): Online reinforcement learning *can* discover behaviors regardless of gain setpoints.

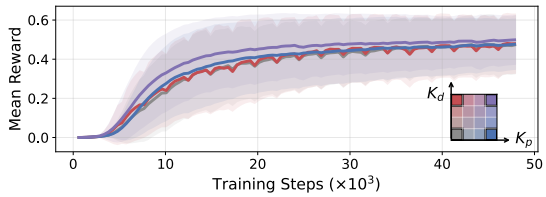
We find that all gain regimes spanning over two orders of magnitude in both $\mathbf{K}_p$ and $\mathbf{K}_d$ *can* yield working controllers given appropriate environment shaping (Table I). Unlike behavior cloning, on-policy RL trains on data generated by its own exploration, which may allow it to learn compensatory



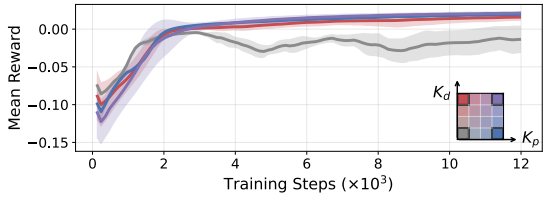
(a) *FR3 Lift-Cube* Hyperparameter Sensitivity



(b) *G1 Track-Velocity* Hyperparameter Sensitivity



(c) *FR3 Lift-Cube* Training Curves



(d) *G1 Track-Velocity* Training Curves

Fig. 10: **RL training across gain regimes.** (a–b) Success rate across the hyperparameter landscape varies among gain settings and tasks; policies with 95%+ success rate (green circles) are found across all conditions. (c–d) Sample efficiency and training stability of PPO is comparable across gain regimes for both tasks.

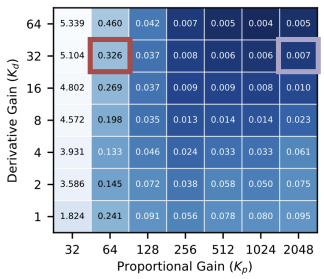
behaviors rather than relying on the controller's error attenuation properties.

> **Result V-B-II** (RL Hyperparameter Sensitivity): The gain setting modulates the hyperparameter optimization landscape, but no regime is consistently easier to optimize.

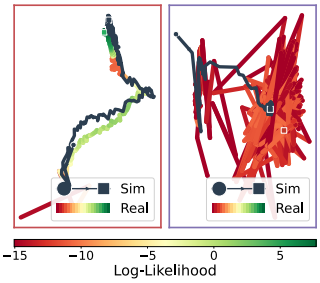
Given that successful policies exist across gain regimes, we ask whether any regime is easier to discover a working policy via hyperparameter optimization. Fig. 10(a–b) visualizes the optimization landscape across environment parameters sampled during hyperparameter optimization for two tasks. While the gain setting clearly modulates the optimization landscape, we do not observe a consistent trend across tasks. This could reflect genuine task-dependence, or it could be an artifact of optimizing only a subset of environment parameters, which represents a low-dimensional slice of a larger optimization space. Resolving this would require larger experiments such as opening up more hyperparameters or leveraging automated reward shaping, which we leave to future work.

> **Result V-B-III** (RL Sample Efficiency): Sample efficiency and training stability are comparable across gain regimes.

We next ask whether any gain regime yields more efficient or stable learning once a working configuration has been identified. We find that training dynamics are comparable across gain regimes for both tasks (Fig. 10(c–d)). The exception is the compliant, underdamped regime on *G1 Track-Velocity*, where only one viable configuration was found; the resulting curve is marginally successful and less smooth, though low seed variance indicates consistent rather than unstable learning.



(a) SysID Modeling Error



(b) Real vs. Sim Rollouts

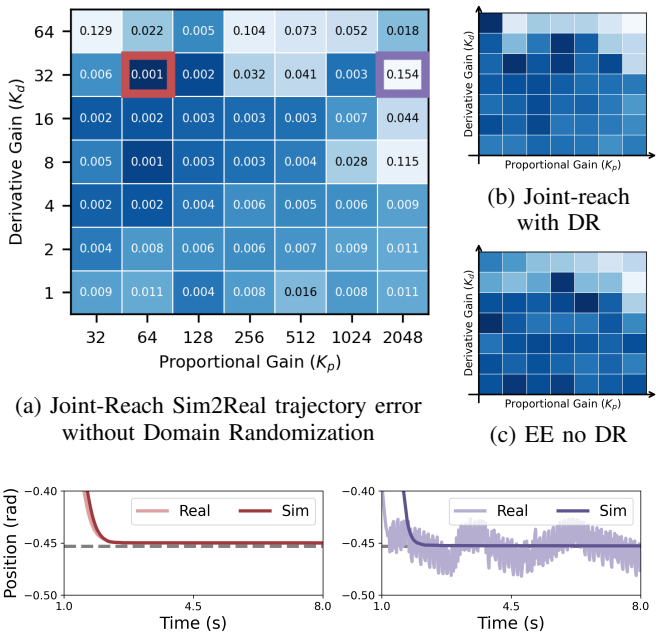
Fig. 11: **Stiff and overdamped gain settings yield lower SysID modeling errors, but exhibit larger closed-loop Sim2Real errors**. Policy observations during closed-loop rollout evolve similarly between sim and real (b-left) for compliant, overdamped gains, but very dissimilarly (b-right) for stiff, overdamped gains.

C. Sim-to-Real

> **Result V-C-I** (System Identification): System identification achieves the lowest modeling error under *stiff* and *overdamped* gains (i.e., upper right region of Fig. 2).

The MSE between simulated and real response curves after system identification, i.e., $\mathcal{S}^\star(\mathbf{K}) = \min_\psi \mathcal{S}(\mathbf{K}, \psi)$ (Eq. 11), is over an order of magnitude lower for the stiff, overdamped regime compared to other gain settings (Fig. 11a).

> **Result V-C-II** (Sim2Real Transferability): Sim2Real transferability, however, is lower with *stiff* and *overdamped* gain setpoints, with its main failure mode being high frequency oscillation.



(a) Joint-Reach Sim2Real trajectory error without Domain Randomization

(b) Joint-reach with DR

(c) EE no DR

(d) Real vs. sim wrist joint trajectories.

Fig. 12: **Stiff and overdamped gain settings reduce sim2real transferability.** The Sim2Real trajectory error (Eq.13) is consistently larger (light blue) in the stiff and overdamped regime (a-c). The primary Sim2Real failure mode is high-frequency oscillation (d).

Trajectory Error. Stiff and overdamped gain settings exhibit the largest sim-to-real trajectory error (Fig. 12). The dominant failure mode is high-frequency oscillation, which persists even with domain randomization (Fig. 12b). Notably, the low-level controller itself is stable under smooth commands; the oscillation only appears during closed-loop policy execution. When we compare the distribution of policy observations between sim and real (Fig. 11b), stiff, overdamped gains produce real-world observations that are highly unlikely under the simulation distribution, whereas compliant, overdamped gains yield closely overlapping distributions.

Statistical Significance. To verify that the stiff-overdamped gain region $\mathcal{G}^{\mathrm{SO}}$ produces significantly larger sim-to-real error, we fit OLS regression on log-transformed trajectory error with $\log_2 \mathbf{K}_\mathrm{p}$ and $\log_2 \mathbf{K}_\mathrm{d}$ as predictors, confirming that both higher $\mathbf{K}_\mathrm{p}$ and higher $\mathbf{K}_\mathrm{d}$ are significant predictors of increased error. We then apply Bonferroni-corrected one-sided Mann-Whitney U tests ($\alpha \approx 0.017$, correcting for 3 conditions) under the null hypothesis

$$\mathcal{H}_0 : \varepsilon(\mathcal{G}^{\mathrm{SO}}) \leq \varepsilon(\mathcal{G} \setminus \mathcal{G}^{\mathrm{SO}}) \tag{15}$$

where ε denotes the sim-to-real trajectory error. $\mathcal{H}_0$ is rejected in all three conditions with $p \ll \alpha$ (Table VIII in Appendix G1).

Discussion. The lower SysID modeling error under stiff, overdamped gains is consistent with these dynamics filtering out nonlinearities that are difficult for idealized simulated actuators to capture: high damping suppresses high-frequency effects such as joint flexibility and transmission dynamics, while high stiffness reduces sensitivity to steady-state errors from imperfect gravity compensation or stiction.

However, the inverse relationship between SysID accuracy and closed-loop transfer quality suggests that gain settings should be evaluated in the context of the full policy loop, not in isolation. We hypothesize that stiff, overdamped controllers amplify small modeling errors because they respond to position and velocity deviations with high torques. When the policy reacts to noise or unmodeled dynamics, the controller aggressively tracks these commands, pushing the system further from states encountered in simulation. Naively choosing the gains that minimize modeling error can therefore paradoxically increase the closed-loop sim-to-real gap.

TABLE I: **RL solution existence across gain regimes.** For each task, we verify that at least one successful policy can be discovered in every gain configuration given appropriate environment shaping. A checkmark indicates that gain regimes corresponding to four corner extremes in Fig. 2 yield working controllers (99%+ success rate). Videos or live policy rollouts of discovered behaviors for each gain settings are available on our project website.

Task / Platform	Existence Proof
FR3 Joint-Reach	✓
FR3 EE-Reach	✓
FR3 Lift-Cube	✓
FR3 Open-Drawer	✓
G1 Track-Velocity	✓
Allegro In-Hand Cube Manipulation	✓
FR3 Nonprehensile Cube Reorientation	✓

VI. Conclusion and Remarks

We have presented a systematic study of how position controller gains shape learning dynamics across three paradigms of modern robot learning. Our findings reveal that gains function not as behavioral parameters, but as an inductive bias that modulates the learning interface between policy and environment. Behavior cloning favors compliant, overdamped regimes; reinforcement learning adapts to any gain setting given compatible hyperparameters; and sim-to-real transfer suffers with stiff, overdamped configurations. These results provide both conceptual clarity and practical guidance for a widely used yet underexplored design decision.

Our framework also raises questions for adjacent areas. Modern humanoid robots increasingly use RL-trained whole-body tracking policies as low-level controllers, analogous to the PD controllers studied here. Yet how their compliance shapes high-level policy learning remains unexplored. Similarly, paradigms that learn manipulation skills from human videos [24, 25] or wearable devices [26] typically treat observed next timestep state as the action label, implicitly assuming perfect target tracking, which our results suggest may be suboptimal for imitation learning. Whether these gain-dependent trends generalize to such cross-embodiment or whole-body control settings remains an open question, and we hope our findings offer a useful lens for investigating these directions.

References

[1] M. Takegaki and S. Arimoto, "A new feedback method for dynamic control of manipulators," 1981.

[2] R. Kelly, "Pd control with desired gravity compensation of robotic manipulators: a review," *The International Journal of Robotics Research*, vol. 16, no. 5, pp. 660–672, 1997.

[3] E. Aljalbout, F. Frank, M. Karl, and P. van der Smagt, "On the role of the action space in robot manipulation learning and sim-to-real transfer," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, p. 5895–5902, Jun. 2024. [Online]. Available: http://dx.doi.org/10.1109/LRA.2024.3398428

[4] D. Kim, G. Berseth, M. Schwartz, and J. Park, "Torque-based deep reinforcement learning for task-and-robot agnostic learning on bipedal robots using sim-to-real transfer," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, p. 6251–6258, Oct. 2023. [Online]. Available: http://dx.doi.org/10.1109/LRA.2023.3304561

[5] J. Eßer, G. B. Margolis, O. Urbann, S. Kerner, and P. Agrawal, "Action space design in reinforcement learning for robot motor skills," in *8th Annual Conference on Robot Learning*, 2024.

[6] Y. Wu, F. Zhao, T. Tao, and A. Ajoudani, "A framework for autonomous impedance regulation of robots based on imitation learning and optimal control," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 127–134, 2021.

[7] K. Kronander and A. Billard, "Learning compliant manipulation through kinesthetic and tactile human-robot interaction," *IEEE Transactions on Haptics*, vol. 7, no. 3, pp. 367–380, 2014.

[8] G. B. Margolis, M. Wang, N. Fey, and P. Agrawal, "Soft-mimic: Learning compliant whole-body control from examples," *arXiv preprint arXiv:2510.17792*, 2025.

[9] N. R. Arachchige, Z. Chen, W. Jung, W. C. Shin, R. Bansal, P. Barroso, Y. H. He, Y. C. Lin, B. Joffe, S. Kousik *et al.*, "Sail: Faster-than-demonstration execution of imitation learning policies," *arXiv preprint arXiv:2506.11948*, 2025.

[10] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," 2017. [Online]. Available: https://arxiv.org/abs/1703.06907

[11] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2018, p. 3803–3810. [Online]. Available: http://dx.doi.org/10.1109/ICRA.2018.8460528

[12] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang, "Solving rubik's cube with a robot hand," 2019. [Online]. Available: https://arxiv.org/abs/1910.07113

[13] F. Muratore, F. Ramos, G. Turk, W. Yu, M. Gienger, and J. Peters, "Robot learning from randomized simulations: A review," 2022. [Online]. Available: https://arxiv.org/abs/2111.00956

[14] S. Gangapurwala, L. Campanaro, and I. Havoutis, "Learning low-frequency motion control for robust and dynamic robot locomotion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5085–5091.

[15] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," *arXiv preprint arXiv:2403.12945*, 2024.

[16] Q. Vuong, S. Levine, H. R. Walke, K. Pertsch, A. Singh, R. Doshi, C. Xu, J. Luo, L. Tan, D. Shah *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," in *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 2023.

[17] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 2003.

[18] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, vol. 44, no. 10-11, pp. 1684–1704, 2025.

[19] Y. Park, G. B. Margolis, and P. Agrawal, "Automatic environment shaping is the next frontier in rl," *arXiv preprint arXiv:2407.16186*, 2024.

[20] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[21] A. Serrano-Muñoz, D. Chrysostomou, S. Bøgh, and N. Arana-Arexolaleiba, "skrl: Modular and flexible library for reinforcement learning," *Journal of Machine Learning Research*, vol. 24, no. 254, pp. 1–9, 2023. [Online]. Available: http://jmlr.org/papers/v24/23-0112.html

[22] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[23] NVIDIA, :, M. Mittal, P. Roth, J. Tigue, A. Richard, O. Zhang, P. Du, A. Serrano-Muñoz, X. Yao, R. Zurbrügg, N. Rudin, L. Wawrzyniak, M. Rakhsha, A. Denzler, E. Heiden, A. Borovicka, O. Ahmed, I. Akinola, A. Anwar, M. T. Carlson, J. Y. Feng, A. Garg, R. Gasoto, L. Gulich, Y. Guo, M. Gussert, A. Hansen, M. Kulkarni, C. Li, W. Liu, V. Makoviychuk, G. Malczyk, H. Mazhar, M. Moghani, A. Murali, M. Noseworthy, A. Poddubny, N. Ratliff, W. Rehberg,

C. Schwarke, R. Singh, J. L. Smith, B. Tang, R. Thaker, M. Trepte, K. V. Wyk, F. Yu, A. Millane, V. Ramasamy, R. Steiner, S. Subramanian, C. Volk, C. Chen, N. Jawale, A. V. Kuruttukulam, M. A. Lin, A. Mandlekar, K. Patzwaldt, J. Welsh, H. Zhao, F. Anes, J.-F. Lafleche, N. Moënne-Loccoz, S. Park, R. Stepinski, D. V. Gelder, C. Amevor, J. Carius, J. Chang, A. H. Chen, P. de Heras Ciechomski, G. Daviet, M. Mohajerani, J. von Muralt, V. Reutskyy, M. Sauter, S. Schirm, E. L. Shi, P. Terdiman, K. Vilella, T. Widmer, G. Yeoman, T. Chen, S. Grizan, C. Li, L. Li, C. Smith, R. Wiltz, K. Alexis, Y. Chang, D. Chu, L. J. Fan, F. Farshidian, A. Handa, S. Huang, M. Hutter, Y. Narang, S. Pouya, S. Sheng, Y. Zhu, M. Macklin, A. Moravanszky, P. Reist, Y. Guo, D. Hoeller, and G. State, "Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning," 2025. [Online]. Available: https://arxiv.org/abs/2511.04831

[24] R.-Z. Qiu, S. Yang, X. Cheng, C. Chawla, J. Li, T. He, G. Yan, D. J. Yoon, R. Hoque, L. Paulsen *et al.*, "Humanoid policy˜ human policy," *arXiv preprint arXiv:2503.13441*, 2025.

[25] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu *et al.*, "Ego4d: Around the world in 3,000 hours of egocentric video," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 18 995–19 012.

[26] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song, "Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots," *arXiv preprint arXiv:2402.10329*, 2024.

[27] Y. Park and P. Agrawal, "Using apple vision pro to train and control robots," 2024. [Online]. Available: https://github.com/Improbable-AI/VisionProTeleop

[28] M. Nomura and M. Shibata, "cmaes : A simple yet practical python library for cma-es," 2024. [Online]. Available: https://arxiv.org/abs/2402.01373

[29] Y. Park, "aiofranka: Asyncio-based franka robot control," 2025. [Online]. Available: https://github.com/Improbable-AI/aiofranka