MANIFOLD GENERATIVE ANNEALING FOR CONTACT CONSTRAINT-AWARE IMPEDANCE CONTROL WITH REINFORCEMENT LEARNING PRIOR

Anonymous authors
Paper under double-blind review

ABSTRACT

Contact-rich robotic tasks require coordinated motion and impedance under uncertain interaction dynamics and physical constraints. While recent methods on learned policies provide useful interaction strategies, they may fail when contact conditions differ from those encountered during training. Model-based generative methods show promising receding-horizon trajectories optimization results, but often lack structured behavioral guidance and refinement that jointly accounts for constraint geometry and actual contact responses. To this end, we propose **Manifold Generative Annealing (MGA)**, a joint model-free and model-based generative framework for motion–impedance optimization. MGA first organizes pretrained reinforcement learning policies into structured sequence proposals, retaining task-specific interaction strategies without committing to their execution. It then evaluates these proposals alongside exploratory samples through model-based rollouts for refinement or separate execution selection. Crucially, realization-aware manifold refinement aligns updates from the refinement bank with both task constraints and locally achievable physical responses, allowing learned experience to guide online adaptation under changing contact conditions. Experiments cover surface scanning, peg insertion, and humanoid pushing, with improved safety success rate of impedance control under nominal and sensing-shift conditions.

1 INTRODUCTION

Contact-rich tasks such as surface scanning, peg insertion, and whole-body pushing require robots to coordinate motion with compliant interaction while respecting physical constraints. Impedance learning and control provide an interface for regulating the relationship between motion and interaction forces (Hogan, 1984; Buchli et al., 2011; Martín-Martín et al., 2019). Selecting these commands remains challenging due to multimodal interaction behaviors, uncertain physical responses, and safety-critical execution, requiring their joint optimization to sustain task progress under changing contact conditions (Abu-Dakka & Saveriano, 2020; Tsuji et al., 2025; Zhang et al., 2025a).

Safe Reinforcement Learning (RL) combines learned task policies with stiffness adaptation, recovery mechanisms, or action safeguards (Zhao et al., 2022; Zhang et al., 2024), providing behavioral structure for contact-rich decisions. Beyond individual policy outputs, the same learned experience can define distributions over candidate behaviors. Model-free diffusion planners provide this generative extension by representing multimodal trajectory distributions, while constrained variants incorporate model-based projections and primal–dual sampling to enforce deployment-time constraints (Römer et al., 2024; Christopher et al., 2024; Zhang et al., 2025b; Blanke et al., 2025). These learned distributions can therefore supply task-informed candidates that reduce unguided search. However, they remain proposals rather than reliable plans because their relevance is inherited from the training distribution and may deteriorate as task or contact conditions change.

Model-Based Diffusion (MBD) (Pan et al., 2024) offers a way to revise sequences through their predicted consequences: it casts trajectory generation as optimization using objective-weighted dynamics rollouts without a learned denoising score. Constrained extensions (He et al., 2026b;a) improve feasibility through Control Barrier Function (CBF) (Ames et al., 2019) or Convex Feasible Set (CFS) (Liu et al., 2018) with adaptive scheduling. Building on this trajectory-optimization view, DIAL-MPC brings diffusion-style annealing to short-horizon model predictive control for online

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

refinement (Xue et al., 2025; Ye et al., 2025). With a limited rollout budget, learned proposals can focus this search on useful contact strategies, and model-based evaluation can reassess them under current conditions. However, weighting promising candidates does not ensure that their combined sequence preserves nonlinear task relationships. Evaluation therefore leaves a second problem: how to correct the update it produces.

Constraint geometry provides local information for correcting such updates. ATACOM constructs safe actions in the tangent space of constraint manifolds (Liu et al., 2022; 2025), while geometry-aware generative optimization uses active constraints to shape refinement and exploration. Geometric impedance control further exploits $\mathrm{SE}(3)$ structure in learned variable-impedance policies for contact-rich manipulation (Seo et al., 2023). Under compliant contact, jamming, or switching, however, displacement commands may produce deformation or force, and stiffness changes the local response. Refinement must therefore account for measured tracking and contact responses when defining the task residual. Shaping the update can reduce the normal response it introduces, but residuals already present or left after a finite step require a further correction. These two operations connect model-based reassessment to continued contact execution.

We propose **M**anifold **G**enerative **A**nnealing (MGA), a short-horizon controller that organizes learned behavioral structure, model-based reassessment, and feedback-conditioned geometric correction in the same motion–impedance sequence space. Rollout weighting determines which candidates inform the update. Geometry shaping adjusts its direction. Local residual correction then revises the updated sequence. The proposal and evaluation stages supply task-informed candidates for refinement or separate execution selection. Our **C**ontributions are three-fold:

- **C1**: We organize model-free learned action policies into structured short-horizon motion–impedance proposals, supplying task-specific interaction behaviors alongside Gaussian exploration in a common sequence space.
- **C2:** We integrate these candidates with model-based generative annealing (online optimization) through common closed-loop evaluation, reassessing their relevance under current task and contact conditions through bank weighting or separate expert selection.
- **C3:** We develop realization-aware manifold refinement that adjusts update directions and corrects remaining task residuals. Contact feedback updates the task references, while response reliability controls the refinement strength. We analyze how the normalized core suppresses encoded normal responses and reduces local residuals.

2 Preliminaries

In this section, we cover the relevant background. Related Works detailed in Appendix A. We use a subscript k for generative steps, a superscript t for physical time, and a subscript m for samples (e.g., $u^t_{k,m}$ denotes sample m's control input at physical time t and generative step k).

2.1 Constrained Lower-Control Sequence Optimization

We consider a robot with state $s^t \in \mathcal{S}$ and generalized lower-control input $u^t \in \mathcal{U}$ over a finite horizon T. Rather than sampling an independent state trajectory, we optimize the finite lower-control sequence $U = (u^0, \dots, u^{T-1}) \in \mathcal{U}^T$. At each step, a task-owned lower-level controller π maps the decision to the physical robot action. Together with the dynamics model f, this controller induces a closed-loop rollout. The resulting constrained lower-control planning problem is

$$\begin{aligned} U^\star \in \arg\min_U \quad & J(U; s^0, c) := \ell_T(s^T; c) + \sum_{t=0}^{T-1} \ell(s^t, u^t; c) \\ \text{s.t.} \quad & a^t = \pi(s^t, u^t; c), \quad s^{t+1} = f(s^t, a^t; c), \quad h(s^t, u^t; c) = 0, \quad g(s^t, u^t; c) \le 0. \end{aligned} \tag{1}$$

Context c contains goals, geometry, contact mode, object or terrain state, and model parameters. The interface covers torque control with π as the identity, impedance control, and whole-body control.

For contact-rich tasks, we instantiate Eq. (1) with the position–stiffness primitive $u^t = (r^t, S^t, \nu^t)$, where r^t is a task, contact, or whole-body reference and ν^t contains feedforward quantities such as desired force, velocity, or footsteps (Hogan, 1984; Buchli et al., 2011; Martín-Martín et al., 2019; Zhang et al., 2024). Here $S^t = (S^t)^\top \in \mathbb{R}^{d\times d}$ is a log-stiffness decision matrix, whereas s^t denotes the robot state. The matrix exponential gives the physical stiffness $K^t = \exp(S^t) \in \mathbb{S}^d_{++}$. Thus positive-definite stiffness is feasible by construction (Arsigny et al., 2007), while h and g impose the remaining constraints. Equalities $h(\cdot)$ encode reliable contact or kinematic consistency. Inequalities

$g(\cdot)$ encode limits such as friction, force, collision, joint range, balance, or tipping margins. To express these horizon constraints as a common zero-level set, we define the residual map

$$C(U, \mu; s^0, c) = \left[\begin{array}{c} h(s^t, u^t; c) \\ g(s^t, u^t; c) + \frac{1}{2}\mu^t \odot \mu^t \end{array} \right]_{t=0}^{T-1}. \tag{2}$$

Thus, $C(U, \mu; s^0, c) = 0$ represents $h = 0$ and $g \leq 0$ through compatible real variables μ, whose squared entries supply nonnegative slack (He et al., 2026b). This defines the hard feasible set in augmented coordinates (Abu-Dakka & Kyrki, 2020; Liu et al., 2022)

$$\mathcal{M}(s^0, c) = \left\{ (U, \mu) \;\middle|\; C(U, \mu; s^0, c) = 0 \right\}. \tag{3}$$

Let $\mathcal{I}_{\text{mode}}$ index discrete contact choices, such as active object faces or stance feet. Each b defines a local branch $\mathcal{M}_b$, giving the union $\mathcal{M}(s^0, c) = \bigcup_{b \in \mathcal{I}_{\text{mode}}} \mathcal{M}_b(s^0, c)$. On a fixed regular branch, let a denote the set of active inequality row indices, where $g_i(s^t, u^t; c) = 0$. With a fixed, $C_a(U)$ collects all equality rows and these active rows after eliminating their slack. Let $J_a(U) = \partial C_a / \partial U$ be its Jacobian through the rollout, with fixed (s^0, c) suppressed. At a feasible point, $J_a \delta U = 0$ defines this branch's tangent space. For a positive-definite sequence metric $G(U)$ with $\|\delta U\|_G^2 = \delta U^\top G \delta U$, the G-orthogonal tangent projector P_a^G is (Dietrich et al., 2015)

$$P_a^G(U) = I - G^{-1} J_a^\top \left(J_a G^{-1} J_a^\top \right)^\dagger J_a. \tag{4}$$

where $\dagger$ denotes the Moore–Penrose pseudoinverse (Penrose, 1955). This local map provides a template for realization-aware refinement, without guaranteeing finite-step feasibility. To combine learned proposals with model-based evaluation, we next express sequence selection as probabilistic inference, assigning greater weight to low-cost, feasible candidates.

2.2 Joint Model-free Model-Based Generative Inference

The constrained cost in Eq. (1) defines the formal model-based compatibility factor

$$\mathcal{L}(U; s^0, c) = \exp[-J(U; s^0, c)/\beta] \, \mathbf{1}\left\{ \exists \mu : \; C(U, \mu; s^0, c) = 0 \right\}, \tag{5}$$

where $\beta > 0$ is a temperature and the indicator enforces the existence of compatible slack variables in Eq. (3). Thus, lower-cost feasible sequences receive higher model-based compatibility.

A model-free component provides a complementary trajectory distribution $Q_\psi(\cdot \mid s^0, c)$. A common instantiation is a state-conditioned RL policy trained with

$$\psi^\star \in \arg\max_{\psi} \quad \mathbb{E}_{\pi_\psi} \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s^t, u^t; c) \right], \tag{6}$$

where $\gamma \in [0, 1)$ and $\mathcal{R}$ is the policy-training reward. Given s^0, the policy induces a sequence distribution through the closed-loop rollout in Eq. (1):

$$u^t \sim \pi_\psi(\cdot \mid o^t, c), \quad U = (u^0, \ldots, u^{T-1}) \sim Q_\psi(\cdot \mid s^0, c), \tag{7}$$

where o^t is the policy observation associated with s^t. More generally, Q_ψ can be induced by any learned policy or trajectory model, including an implicit or deterministic generator. Its training objective need not coincide with the test-time planning objective: the model-free component supplies behavioral structure, whereas the model-based factor supplies test-time evaluation.

Joint model-free model-based inference reweights this learned trajectory distribution using model-based evidence (Jung et al., 2025). Directly using $\mathcal{L}$ gives a zero normalizer when the equality constraints define a lower-dimensional feasible set and Q_ψ has an ambient density. To retain nonzero weights for mildly infeasible candidates, we use the softened cost $\widetilde{J}(U; s^0, c) = J(U; s^0, c) + \mathcal{V}(U; s^0, c)$, where the scalar penalty $\mathcal{V} \geq 0$ measures task-selected constraint violations along the rollout. This penalty evaluates candidates without enforcing hard feasibility or replacing geometric refinement. Following Boltzmann distribution (Watson et al., 2020; Watson & Peters, 2023), the resulting joint-inference target is

$$P_\beta(dU \mid s^0, c) \propto \exp[-\widetilde{J}(U; s^0, c)/\beta] \, Q_\psi(dU \mid s^0, c), \tag{8}$$

assuming a finite, nonzero normalizer. This is a prior-weighted surrogate for Eq. (1). For samples from Q_ψ, weighting requires only rollout costs, not a prior density or score. Reweighting cannot expand the support of Q_ψ, leaving a deterministic horizon unchanged. MGA therefore complements learned proposals with local Gaussian exploration. We next introduce a generative sampling view that separates clean-sequence estimation from the reverse transport that updates the sequence.

2.3 Generative Models for Trajectory Sampling

Take P_β in Eq. (8) as the clean sequence measure. Generative sampling refines a noisy sequence U_K toward a clean sequence U_0. In normalized control coordinates, we define a Gaussian probability path by $U_k = \sqrt{\bar{\alpha}_k} U_0 + \sqrt{1-\bar{\alpha}_k}\,\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$, $\bar{\alpha}_k = \prod_{j=1}^{k} \alpha_j$ with $\alpha_j \in (0,1)$, $\bar{\alpha}_0 = 1$, and $\bar{\alpha}_K \approx 0$. For $k \geq 1$, the forward kernel $q_k(U_k \mid U_0) = \mathcal{N}(U_k; \sqrt{\bar{\alpha}_k} U_0, (1-\bar{\alpha}_k) I)$ is the conditional density of the noisy sequence given the clean one. The noisy marginal is $p_k(U_k) = \int q_k(U_k \mid U_0) P_\beta(dU_0)$. The clean conditional law is

$$P_{0|k}(dU_0 \mid U_k) \propto q_k(U_k \mid U_0) \exp[-\widetilde{J}(U_0)/\beta]\, Q_\psi(dU_0). \tag{9}$$

Let $\bar{U}_k := \mathbb{E}_{P_{0|k}}[U_0 \mid U_k]$ be the conditional clean mean. The Gaussian Tweedie identity (Efron, 2011) relates it to the noisy score through

$$\mathcal{S}_k(U_k; \widehat{U}_0) = -\frac{U_k}{1-\bar{\alpha}_k} + \frac{\sqrt{\bar{\alpha}_k}}{1-\bar{\alpha}_k} \widehat{U}_0, \tag{10}$$

where $\widehat{U}_0$ is a generic estimate of the clean sequence U_0. The exact score satisfies $\nabla_{U_k} \log p_k(U_k) = \mathcal{S}_k(U_k; \bar{U}_k)$. A learned denoiser can approximate $\bar{U}_k$, while model-based methods use rollout evaluations to construct a clean estimate.

For the Gaussian-only diffusion special case, a flat reference replaces Q_ψ. Normalizing $q_k(U_k \mid U_0)$ over U_0 gives the proposal used in model-based sampling (Pan et al., 2024; Xue et al., 2025):

$$\widetilde{U}_{k,m} \sim \Omega_k(U_k) := \mathcal{N}\left(\frac{U_k}{\sqrt{\bar{\alpha}_k}}, \left(\frac{1}{\bar{\alpha}_k} - 1\right) I\right), \quad \widehat{\bar{U}}_k = \frac{\sum_{m=1}^{M} w_{k,m} \widetilde{U}_{k,m}}{\sum_{m=1}^{M} w_{k,m}}. \tag{11}$$

where $w_{k,m} = e^{-\widetilde{J}(\widetilde{U}_{k,m})/\beta}$. Each candidate is evaluated through Eq. (1). These cost-only weights rely on the flat reference. Arbitrary learned proposals do not yield the same cancellation.

The clean estimate $\widehat{\bar{U}}_k$, or its corresponding score, can drive different reverse transports, including stochastic diffusion, DDIM, and probability-flow updates (Song et al., 2021a;b). We write these as

$$U_{k-1} = \mathcal{T}_k(U_k, \widehat{\bar{U}}_k, z_k), \quad z_k \sim \mathcal{N}(0, I), \tag{12}$$

where $\mathcal{T}_k$ specifies the transport (DDPM, DDIM, Flow-like), and deterministic updates omit z_k. The question now becomes how to construct structured proposals and refine their clean moment using realization geometry before applying the chosen transport.

3 Methodology

MGA couples three steps in motion–impedance sequence space. (1) Learned policies and Gaussian exploration supply candidate behaviors. (2) Common closed-loop rollouts reassess candidates for refinement or separate execution selection. (3) Geometric refinement coordinates the weighted bank update with task residuals under current feedback. Directional shaping reduces the update's encoded normal response. Local residual correction (LRC) then corrects residuals remaining afterward.

3.1 RL Prior-guided Structural Proposals

The policy supplies short-horizon interaction behaviors. Gaussian proposals explore local alternatives. At reverse step k, they are centered on the clean sequence $\check{U}_k := U_k/\sqrt{\bar{\alpha}_k}$ under the Gaussian path in Section 2.3. Policy rollouts follow Q_ψ. In solver-node coordinates U, rollouts and residuals include fixed interpolation (Appendix H). The refinement bank is

$$\mathcal{Q}_k = \{\widetilde{U}^{\mathrm{g}}_{k,m}\}_{m=1}^{M_{\mathrm{g}}} \uplus \{\widetilde{U}^{\psi}_{k,m}\}_{m=1}^{M_\psi}, \quad M_{\mathrm{g}} + M_\psi = M, \quad M_{\mathrm{g}} > 0. \tag{13}$$

Here M counts refinement candidates. Replacement exchanges Gaussian for learned slots at fixed M. Additive sets $M_\psi = 0$ and compares separate policy horizons with the refined plan at execution selection. Learned horizons are drawn once per solve. Replacement slots are reused across reverse steps, while Gaussian slots are resampled. The anchor set $\mathcal{A}^{\mathrm{anc}}_k$ contains $\check{U}_k$ and, when used, a deterministic policy rollout. Appending them gives $N_k = M + |\mathcal{A}^{\mathrm{anc}}_k|$ slots $\widetilde{U}_{k,m}$.

Model-based weighting. The candidate sources suggest behaviors, but their usefulness depends on current contact conditions. We evaluate bank candidates through the same closed-loop model in Eq. (1). The softened objective $\widetilde{J}$ weights them into a clean sequence estimate:

$$\bar{w}_{k,m} = \mathrm{softmax}_m\left(-\frac{\widetilde{J}(\widetilde{U}_{k,m})}{\beta_k}\right), \quad \widehat{\bar{U}}_k = \sum_{m=1}^{N_k} \bar{w}_{k,m} \widetilde{U}_{k,m}. \tag{14}$$

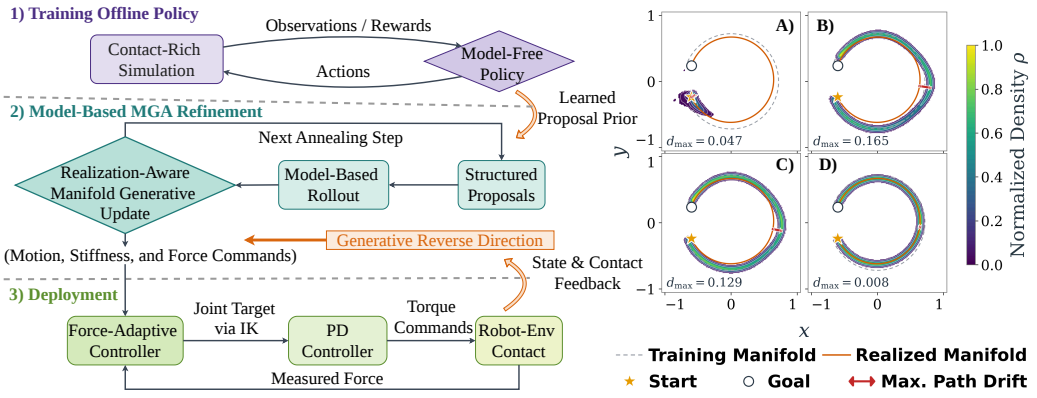


Figure 1: Left: Overview of MGA: (1) offline policy training, (2) model-based refinement, and (3) force-adaptive deployment with feedback. Right: an open-circle start-to-goal toy: (A) Gaussian-initialized refinement, (B) RL prior, (C) MGA w/o LRC, and (D) MGA. The prior supplies path structure. LRC reduces maximum median-path drift: 0.165 (B), 0.129 (C), 0.008 (D). Arrows mark full-path maxima of distance to the realized manifold. C/D differ only in LRC.

Here $\beta_k > 0$ is the cost-scaled temperature. Rollout costs determine each candidate's contribution without requiring a learned density or score. This finite-bank weighting follows the principle of Eq. (8), without sampling exactly from P_β. The proposed update $d_k := \widehat{\bar{U}}_k - \check{U}_k$ moves the current clean-coordinate sequence toward the weighted estimate. Even low-cost candidates can yield a weighted combination that disrupts nonlinear task relationships. We therefore shape this update's direction and correct the resulting sequence's task residual.

3.2 Realization-Aware Manifold Refinement

The residual $C^{\mathrm{real}}(U)$ measures a sequence's deviation from task references. Where enabled, tracking and contact feedback adjust these references using a local response estimate. This correction Δ^{real} remains fixed during the solve. The local zero-level set is $\mathcal{M}^{\mathrm{real}} := \{U : C^{\mathrm{real}}(U) = 0\}$. Setting $\Delta^{\mathrm{real}} = 0$ recovers the nominal task references. This residual encodes selected task relationships, distinct from the full hard-feasibility residual in Eq. (2). Task-specific definitions and feedback corrections are given in Appendix F.

We first shape the proposed update d_k. Let $\mathsf{J}(U) := \partial C^{\mathrm{real}}(U)/\partial U$ be the residual Jacobian. We construct A_k from node blocks of the residual-energy gradient (Appendix H), so $A_k d_k$ measures the update along encoded normal directions. For a metric $G_k \succ 0$ and regularization $\lambda_k > 0$, the filter below gives the shaped update $\mathsf{M}_k d_k$, the updated clean estimate, and its reverse field:

$$\begin{aligned} \mathsf{M}_k &:= I - G_k^{-1} A_k^\top \left(\lambda_k I + A_k G_k^{-1} A_k^\top\right)^{-1} A_k, \\ \bar{U}_{\mathcal{M},k} &:= \check{U}_k + \mathsf{M}_k d_k, \quad \widehat{S}_{\mathcal{M},k} := \mathcal{S}_k(U_k; \bar{U}_{\mathcal{M},k}). \end{aligned} \tag{15}$$

The filter preserves directions z with $A_k z = 0$ and attenuates the encoded response $A_k d_k$. Smaller λ_k suppresses the encoded responses more strongly, while larger values retain more of d_k. At a regular point of $\mathcal{M}^{\mathrm{real}}$, choosing $A_k = \mathsf{J}(\check{U}_k)$ and taking $\lambda_k \downarrow 0$ recovers the metric tangent map in Eq. (4). We use $G_k = I$ in normalized motion–impedance coordinates. Log-stiffness coordinates preserve positive definiteness, and rollout evaluation accounts for motion–impedance coupling. Task-dependent gain sets the refinement strength. Where enabled, response reliability also modulates the change to the rollout-weighted update (Appendix H).

We next carry the shaped estimate to reverse step $k-1$ using the MBD-style transport in Eq. (12):

$$U_{k-1}^{\mathrm{raw}} = \frac{1}{\sqrt{\alpha_k}} \left(U_k + (1 - \bar{\alpha}_k)\widehat{S}_{\mathcal{M},k}\right), \quad \frac{U_{k-1}^{\mathrm{raw}}}{\sqrt{\bar{\alpha}_{k-1}}} = \bar{U}_{\mathcal{M},k}. \tag{16}$$

The identity on the right follows from $\bar{\alpha}_k = \alpha_k \bar{\alpha}_{k-1}$: transport changes the reverse-step coordinates while preserving the shaped clean estimate.

Local Residual Correction (LRC). Shaping controls the update direction, but existing residuals and finite-step errors can remain. LRC therefore corrects the transported clean sequence $V_k :=$

$U_{k-1}^{\text{raw}}/\sqrt{\bar{\alpha}_{k-1}} = \bar{U}_{\mathcal{M},k}$ using a damped Jacobian step:

$$V_k^+ = V_k - \eta \mathsf{J}(V_k)^\top \left[\mathsf{J}(V_k)\mathsf{J}(V_k)^\top + \varepsilon I\right]^{-1} C^{\text{real}}(V_k), \quad U_{k-1} = \sqrt{\bar{\alpha}_{k-1}} V_k^+, \tag{17}$$

Here $\eta > 0$ sets the correction gain and $\varepsilon > 0$ regularizes the linear solve. The Jacobian-based term maps the remaining residual $C^{\text{real}}(V_k)$ into a sequence correction. LRC evaluates the full Jacobian at V_k, whereas shaping uses A_k at $\check{U}_k$. This local correction is neither an exact projection nor a hard-feasibility guarantee. Implementation details appear in Appendix H.

3.3 Receding-Horizon Execution

Algorithm 1 summarizes one replanning cycle. MGA initializes from the shifted plan, optionally mixed with a deterministic policy rollout, and fixes Δ^{real}. Each reverse step evaluates the refinement bank. The resulting d_k undergoes directional shaping, transport, and LRC. The algorithm presents the normalized geometric core. Gain, reliability modulation, and localization are specified in Appendix H. After refinement, execution acceptance, when enabled, re-evaluates the candidate against the task-defined incumbent. Additive experts then compete through separate evaluation and acceptance. The lower-level controller executes the first primitive, and the remaining horizon is shifted for the next warm start.

Algorithm 1 Manifold Generative Annealing (MGA)

Require: State s^0, task c, learned proposal Q_ψ, shifted plan U^{shift}, bank size M.
1: Initialize U^{init} (Appendix H) and set $U_K \leftarrow \sqrt{\bar{\alpha}_K} U^{\text{init}}$.
2: Freeze Δ^{real} and draw learned slots for replacement.
3: **for** $k = K, \dots, 1$ **do**
4: Set $\check{U}_k \leftarrow U_k/\sqrt{\bar{\alpha}_k}$ and draw Gaussian slots.
5: Form $\mathcal{Q}_k$ with learned slots only for replacement (13), then append anchors.
6: Evaluate model-based rollout costs and compute the clean estimate by (14).
7: Build A_k and shape the clean displacement d_k by (15).
8: Transport to U_{k-1}^{raw} by (16).
9: Correct the transported clean sequence and rescale to U_{k-1} by (17).
10: **end for**
11: Select the plan by task acceptance of U_0 against the incumbent, when enabled.
12: Draw and evaluate additive experts, if used, then test the best admissible expert against the selected plan.
13: Execute its first primitive and shift the remaining horizon.
14: Update state and contact feedback for the next replan.
Ensure: Selected control sequence and next warm start.

4 Theoretical Analysis

We summarize the main theoretical properties of MGA and defer detailed proofs to Appendix B. The analysis isolates the effects of model-based reweighting, geometric refinement, and local residual correction. We condition on the evaluated batch at reverse step k, with finite costs and $\beta_k > 0$, and fix A_k, $G_k \succ 0$, and $\lambda_k > 0$. The results concern the normalized update before task-dependent modulation. We use $\|x\|_{G_k}^2 = x^\top G_k x$.

Lemma 1 (Finite-bank Model-based Reweighting). Let $j_m := \widetilde{J}(\widetilde{U}_{k,m})$ and let Δ_{N_k} be the probability simplex over the N_k evaluated candidates. The weights in Eq. (14) uniquely solve

$$\bar{w}_k = \arg\min_{w \in \Delta_{N_k}} \left\{ \sum_{m=1}^{N_k} w_m j_m + \beta_k D_{\text{KL}}\left(w \middle\| \frac{\mathbf{1}}{N_k}\right) \right\}. \tag{18}$$

Moreover, $\sum_m \bar{w}_{k,m} j_m \le \min_m j_m + \beta_k \log N_k$. Thus model-based weighting concentrates the evaluated distribution toward lower-cost candidates, regardless of their proposal source. This statement concerns weighted candidate costs, not the cost or feasibility of their ambient mean.

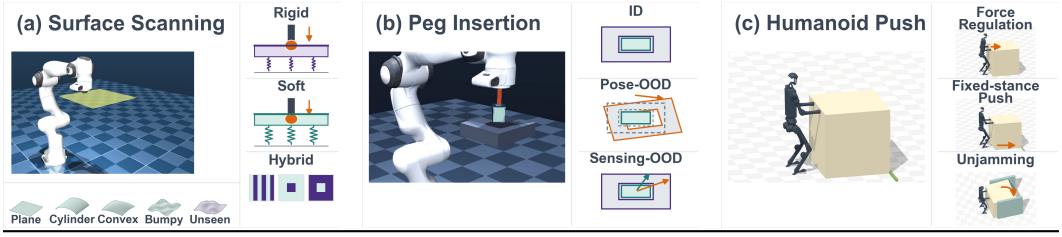
Lemma 2 (Regularized Geometric Refinement). For any clean displacement d, let $v = \mathsf{M}_k d$ and $H_k = A_k G_k^{-1} A_k^\top$. Then

$$v = \arg\min_z \left\{ \frac{1}{2}\|z - d\|_{G_k}^2 + \frac{1}{2\lambda_k}\|A_k z\|_2^2 \right\}, \quad A_k v = \lambda_k(\lambda_k I + H_k)^{-1} A_k d. \tag{19}$$

The minimizer is unique, $\|v\|_{G_k} \le \|d\|_{G_k}$, and the G_k-orthogonal component in $\ker A_k$ is preserved. Every nonzero encoded normal response is strictly attenuated. Equation (16) transports this shaped displacement directly into the next clean iterate.

Theorem 1 (Local Residual Control). Suppose $C^{\text{real}}(U) = \mathsf{J}U - b$ on a fixed smooth affine branch containing $\check{U}_k$, $V_k = \check{U}_k + \mathsf{M}_k d_k$, and the corrected sequence V_k^+. Let $\Pi = \mathsf{J}\mathsf{J}^\dagger$ and

Figure 2: The environments are (a) Surface Scanning over rigid (purple) and soft (green) surfaces with varied geometries and hybrid hard–soft patterns, (b) Peg Insertion into a fixed socket under nominal conditions (ID), perturbed socket poses (Pose-OOD), and biased force/torque sensing (Sensing-OOD), and (c) Humanoid Push for constant 15-N force regulation, fixed-stance pushing to the green line, and box unjamming with yaw correction.



	Environment	Metric	ISSA	ATACOM	MPPI	DIAL	PegasusFlow	MGA	w/o RL prior	w/o LRC
Surface Scanning	Hard	S ↑	5.0±10.0	25.0±0.0	22.5±20.8	62.5±20.2	5.0±10.0	**100.0**±0.0	**100.0**±0.0	67.5±19.5
		F ↓	0.596±0.050	0.533±0.015	1.029±0.094	0.848±0.122	1.301±0.237	0.498±0.037	**0.486**±0.017	0.521±0.039
	Soft	S ↑	2.5±7.5	50.0±0.0	20.0±15.0	77.5±17.5	30.0±24.5	**100.0**±0.0	**100.0**±0.0	70.0±15.0
		F ↓	0.596±0.051	0.470±0.011	0.973±0.059	0.762±0.098	1.094±0.202	**0.466**±0.024	0.488±0.031	0.498±0.017
	Hybrid	S ↑	0.0±0.0	66.7±0.0	30.0±10.0	56.7±21.3	20.0±16.3	**100.0**±0.0	96.7±10.0	56.7±21.3
		F ↓	0.698±0.037	**0.606**±0.000	0.879±0.034	0.789±0.031	0.862±0.032	0.658±0.016	0.738±0.016	0.702±0.020
	Unseen	S ↑	0.0±0.0	50.0±0.0	45.0±26.9	65.0±32.0	15.0±22.9	50.0±0.0	55.0±15.0	**75.0**±25.0
		F ↓	0.555±0.025	**0.417**±0.016	0.827±0.067	0.754±0.149	1.018±0.353	0.429±0.023	0.433±0.018	0.502±0.030
Peg Insertion	ID	S ↑	0.0±0.0	0.0±0.0	60.0±49.0	70.0±45.8	60.0±49.0	**100.0**±0.0	90.0±30.0	50.0±50.0
		F ↓	0.723±0.102	1.079±0.000	**0.162**±0.221	0.620±0.186	0.670±0.141	0.775±0.010	0.733±0.093	0.449±0.289
	Pose-OOD	S ↑	0.0±0.0	50.0±50.0	30.0±45.8	40.0±49.0	**80.0**±40.0	**80.0**±40.0	50.0±50.0	10.0±30.0
		F ↓	1.203±0.347	0.916±0.053	**0.366**±0.248	0.740±0.221	0.630±0.227	0.893±0.093	0.769±0.156	0.477±0.365
	Sensing-OOD	S ↑	0.0±0.0	0.0±0.0	80.0±40.0	60.0±49.0	90.0±30.0	**100.0**±0.0	**100.0**±0.0	30.0±45.8
		F ↓	0.660±0.053	0.969±0.000	**0.204**±0.273	0.601±0.228	0.667±0.089	0.774±0.011	0.775±0.041	0.435±0.300
Humanoid Push	Force Regulation	S ↑	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0	90.0±30.0	70.0±45.8	**100.0**±0.0	**100.0**±0.0	90.0±30.0
		F ↓	0.271±0.000	0.270±0.000	0.288±0.004	0.357±0.028	0.335±0.016	**0.262**±0.005	0.280±0.000	0.263±0.004
	Fixed-stance Push	S ↑	0.0±0.0	0.0±0.0	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0
		F ↓	0.972±0.000	1.057±0.000	0.557±0.019	0.550±0.022	0.559±0.016	0.503±0.038	**0.492**±0.005	0.538±0.010
	Unjamming	S ↑	0.0±0.0	0.0±0.0	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0	80.0±40.0	30.0±45.8
		F ↓	1.458±0.159	1.601±0.000	**0.798**±0.006	0.805±0.008	0.807±0.012	0.845±0.026	0.742±0.005†	0.862±0.014‡

Table 1: Quantitative results of safe-success rate (**S**) (%), and normalized contact-load tail risk (**F**) (nCVaR_{95}) across three contact-rich tasks. Table entries report mean and standard deviation over 10 seeds. Bold and underlining indicate the best and second-best displayed values, including ties.

$E(U) = \|\Pi C^{\mathrm{real}}(U)\|_2$. For $0 < \eta \leq 1$, $\varepsilon > 0$, and $\mathrm{rank}(\mathsf{J}) > 0$, Eq. (17) satisfies

$$E(V_k^+) \leq \rho \left\|\Pi C^{\mathrm{real}}(\check{U}_k) + \mathsf{J}\mathsf{M}_k d_k\right\|_2 = \rho E(V_k), \quad \rho = 1 - \eta \frac{\sigma^2_{\min,+}(\mathsf{J})}{\sigma^2_{\min,+}(\mathsf{J}) + \varepsilon} < 1, \tag{20}$$

where $\sigma_{\min,+}$ is the smallest positive singular value. The uncorrectable component remains $-(I - \Pi)b$. It vanishes when $b \in \mathrm{range}(\mathsf{J})$. Lemma 2 controls the encoded response of the proposed displacement, whereas Theorem 1 reduces the removable chart residual after transport (Appendix B).

5 Experimental Analysis

We evaluate our MGA and its ablation variants through three questions: (**RQ1:**) Can MGA maintain contact-valid progress while controlling contact force and surface deformation under contact-response mismatch? (**RQ2:**) How do learned policy priors complement model-based refinement in achieving success under contact and sensing mismatch? (**RQ3:**) Can MGA extend from fixed-base manipulation to whole-body pushing, where balance, contact, and object motion are coupled?

Experimental Setup. We evaluate **MGA** on three tasks. (1) In Surface Scanning, a robot arm follows a surface path while regulating contact force across rigid and soft surfaces, unseen geometric shapes, and hybrid hard–soft regions. (2) In Peg Insertion, a robot arm inserts a rectangular peg into a fixed socket under nominal conditions (ID), perturbed socket poses (Pose-OOD), and biased force/torque sensing (Sensing-OOD). (3) In Humanoid Push, a Unitree H1 humanoid coordinates

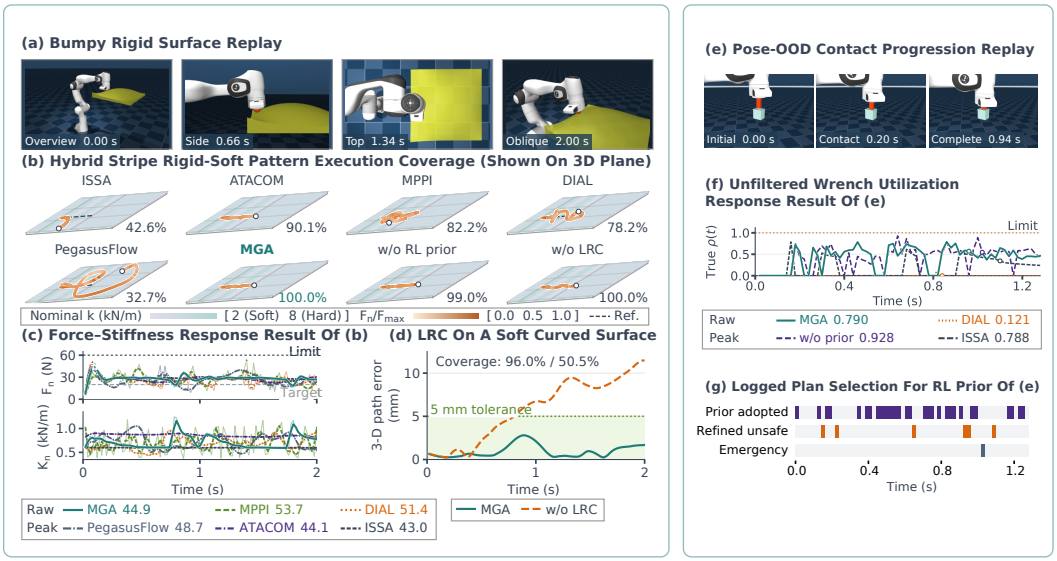


Figure 3: Left: Scanning panels show a (a) bumpy rigid surface replay, (b) hybrid stripe rigid-soft pattern execution coverage and their (c) force–stiffness response, and (d) compares MGA and w/o LRC on a soft curved surface. Right panels show Pose-OOD Peg Insertion (e) contact progression replay, (f) unfiltered wrench utilization, and (g) logged plan selection for RL prior.

balance, hand contact, and box motion during 15-N force regulation, nominal fixed-stance pushing to the line (green), and unjamming the box with yaw correction. The environments are shown in Fig. 2. Environment details appear in Appendix C.

Baselines. We compare **MGA** with five baselines. (1) ISSA (Zhao et al., 2022) filters learned policy actions through an Implicit Safe-Set. (2) ATACOM (Liu et al., 2022) constructs learned actions on an augmented constraint manifold. (3) MPPI (Williams et al., 2018) uses reward-weighted trajectory sampling. (4) DIAL (Xue et al., 2025) performs diffusion-style MPC annealing. (5) PegasusFlow (Ye et al., 2025) extends to rolling denoising within the annealing. Two ablations complement these baselines: **w/o RL prior** removes learned proposals, and **w/o local residual correction (LRC)** removes LRC. Details appear in Appendix D.

5.1 Experimental Results

Performance across Contact-Rich Tasks. As shown in Table 1, on rigid, soft and hybrid surfaces, MGA achieves 100% safe success with lower force tails than MPPI, DIAL, and PegasusFlow, indicating that its coverage is not obtained by pushing harder through contact. In Peg Insertion, MGA achieves 100%, 80%, and 100% safe success under ID, Pose-OOD, and Sensing-OOD, respectively. It completes insertion in all 30 trials, but two Pose-OOD trials exceed the force limit. Its higher safe success relative to the tested action-level methods supports the practical value of planning and correcting contact sequences. In Humanoid Push, MGA completes all three tasks safely and lowers tail loads relative to the sampling-based planners during force regulation and fixed-stance pushing. Together, these results support contact-conditioned sequence refinement for preserving task progress while managing interaction loads across the robots and contact regimes.

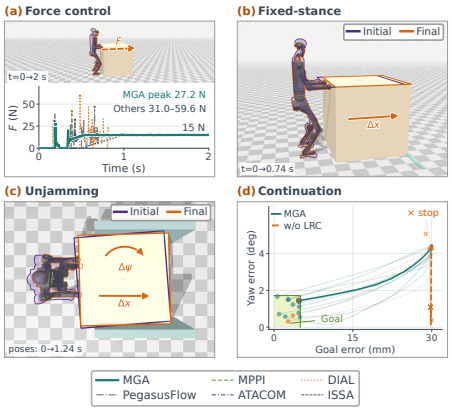


Figure 4: Panels (a–c) show force regulation, pushing, and unjamming. Panel (d) shows 10 paired position–yaw trajectories. Crosses mark safety-triggered stops. Force transients are retained in (a).

5.2 Discussion

Can MGA Preserve Contact-Valid Progress? (RQ1) In Fig. 3(b–c), MGA covers the entire hybrid path while adapting its normal stiffness and keeping contact force below the limit. MPPI, DIAL, and PegasusFlow exhibit larger force peaks but lower coverage, showing that stronger interaction

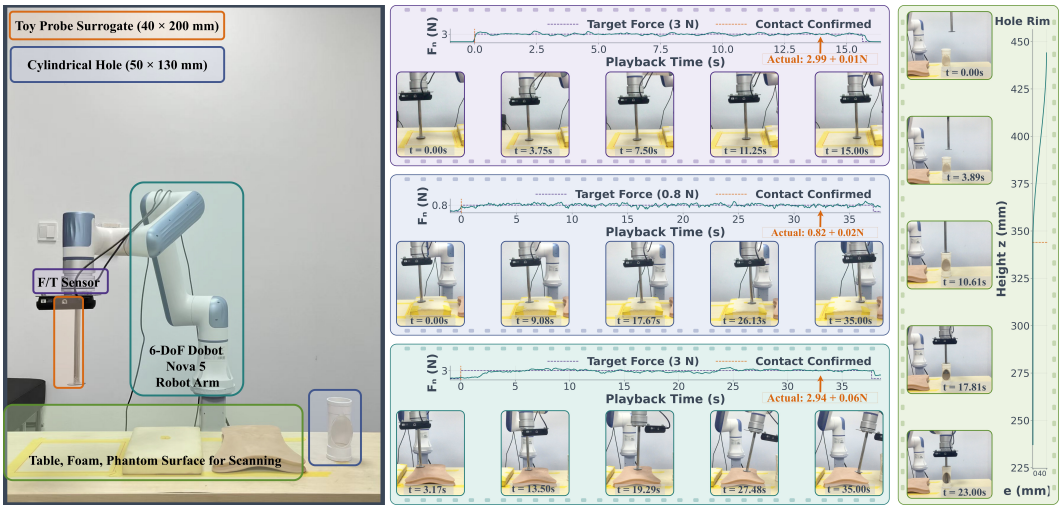


Figure 5: Left shows the hardware setup. Middle shows table, foam, and phantom scanning from top to bottom, with force references of 3, 0.8, and 3 N (Actual: 2.99 ± 0.01, 0.82 ± 0.02, 2.94 ± 0.06 N, respectively). Right plots lateral offset e (mm) against base frame height z (mm) during insertion.

does not necessarily yield useful progress. On the curved soft surface in panel (d), removing LRC causes the executed path to exceed the 5-mm tolerance and reduces coverage from 96.0% to 50.5%, whereas MGA remains within the tolerance. These matched executions illustrate why geometric shaping alone is insufficient: correcting residuals left by finite updates helps preserve contact-valid motion under changing surface response.

How Do Learned Priors Complement Model-Based Refinement? (RQ2) RL-based methods acquire useful insertion behavior, but task competence alone does not ensure safe execution: ISSA and ATACOM complete $29/30$ and $30/30$ trials, yet only $0/30$ and $5/30$ are safe (Fig. 10). MGA instead reassesses learned horizons separately and compares them with geometrically refined Gaussian sequences under the additive protocol. Across all three Peg Insertion settings, this combination yields $28/30$ safe completions, versus $17/30$ each for MPPI and DIAL and $23/30$ for PegasusFlow (Table 1). Removing the prior reduces MGA to $24/30$, supporting the value of additional learned candidates under this protocol. Together, these comparisons support combining learned behavioral structure with model-based reassessment and sequence refinement.

Can MGA Extend to Whole-Body Pushing? (RQ3) MGA safely completes all three humanoid tasks (Table 1), supporting extension beyond fixed-base manipulation. Fig. 4(d) shows all ten MGA trajectories reaching the position–yaw goal region, whereas w/o LRC completes only three, with seven trials ending when no candidate passes revalidation. Thus, contact-conditioned sequence correction sustains useful continuations under coupled balance, hand contact, and object motion, while retaining the same execution checks.

5.3 Real-world surface scanning and insertion

We evaluate the model-based component of MGA on a 6-DoF Dobot Nova 5 with wrist force/torque (F/T) sensing (Fig. 5). The robot scans a rigid table, compliant foam, and a curved anatomical phantom while regulating normal force. Foam requires repeated contact recovery, whereas the phantom requires adapting tool height and orientation along the surface. An additional hole-insertion experiment requires alignment before entry and monitored advancement. These observations support its practical use across different contact conditions. Details appear in Appendix E.

6 Conclusion

We presented MGA, an online motion–impedance optimization method that combines RL policies as structured proposals, model-based reassessment, and feedback-conditioned geometric correction. Rollout weighting identifies useful candidate contributions, geometric shaping adjusts the resulting update, and LRC addresses remaining task residuals. The experiments support the complete refinement method, with learned proposals improving insertion under socket-pose shifts and LRC supporting safe completion across all three insertion conditions. Future work will combine richer proposals from perception models with task-level planning to enable longer-horizon force-control tasks and extend MGA to coordinated multi-robot complex physical interaction settings.

7 AI USE DISCLOSURE

Large Language Models (LLMs) were employed during the preparation of this manuscript to support both the research and writing processes. Specifically, we used LLMs to assist with research ideation and aspects of research execution, including discussions of potential research directions and support for parts of the research workflow. We also used LLMs to help draft portions of the manuscript and to improve the clarity, readability, grammar, and overall presentation of the text.

The use of LLMs was primarily supportive rather than autonomous. Research ideas, methodological decisions, analyses, and final interpretations were reviewed and determined by the authors. Any LLM-assisted research suggestions or drafted material were critically examined and revised where necessary, and the scientific content and experimental results presented in the paper were independently verified by the authors.

The authors take full responsibility for the final manuscript and all of its contents, including materials produced with the assistance of LLMs. All AI-assisted content was reviewed by the authors to ensure its accuracy, consistency with the actual research, and compliance with applicable ethical and academic standards.

REFERENCES

Fares J. Abu-Dakka and Ville Kyrki. Geometry-aware dynamic movement primitives. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4421–4426, 2020. doi: 10.1109/ICRA40945.2020.9196952.

Fares J Abu-Dakka and Matteo Saveriano. Variable impedance control and learning—a review. *Frontiers in Robotics and AI*, 7:590681, 2020.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 22–31. PMLR, 2017.

Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, pp. 3420–3431. Ieee, 2019.

Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Log-euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 56(2):411–421, 2006.

Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 29(1):328–347, 2007. doi: 10.1137/050637996.

Matthieu Blanke, Yongquan Qu, Sara Shamekh, and Pierre Gentine. Strictly constrained generative modeling via split augmented langevin sampling, 2025. URL https://arxiv.org/abs/2505.18017.

Jonas Buchli, Freek Stulp, Evangelos Theodorou, and Stefan Schaal. Learning variable impedance control. *The International Journal of Robotics Research*, 30(7):820–833, 2011.

Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. In *International Conference on Learning Representations*, volume 2024, pp. 47922–47945, 2024.

Ching-An Cheng, Mustafa Mukadam, Jan Issac, Stan Birchfield, Dieter Fox, Byron Boots, and Nathan Ratliff. Rmpflow: A geometric framework for generation of multitask motion policies. *IEEE Transactions on Automation Science and Engineering*, 18(3):968–987, 2021.

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

Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. Lyapunov-based safe policy optimization for continuous control. *arXiv preprint arXiv:1901.10031*, 2019.

Jacob K. Christopher, Stephen Baek, and Ferdinando Fioretto. Constrained synthesis with projected diffusion models. *Advances in Neural Information Processing Systems*, 37:89307–89333, 2024.

Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018.

Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modelling. *Advances in neural information processing systems*, 35:2406–2422, 2022.

Akua K Dickson, Juan C Pacheco Garcia, Meredith L Anderson, Ran Jing, Sarah Alizadeh-Shabdiz, Audrey X Wang, Charles DeLorey, Zach J Patterson, and Andrew P Sabelhaus. Safe autonomous environmental contact for soft robots using control barrier functions. *IEEE Robotics and Automation Letters*, 2025.

Alexander Dietrich, Christian Ott, and Alin Albu-Schäffer. An overview of null space projections for redundant, torque-controlled robots. *The International Journal of Robotics Research*, 34(11): 1385–1400, 2015.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. doi: 10.1198/jasa.2011.tm11181.

Jonas Günster, Puze Liu, Jan Peters, and Davide Tateo. Handling long-term safety and uncertainty in safe reinforcement learning. *arXiv preprint arXiv:2409.12045*, 2024.

Zhilin He, Bowei Li, Jianlin Dou, Yuner Zhang, and Changliu Liu. Motion planning with model-based diffusion via constraint optimization and adaptive scheduling. *arXiv preprint arXiv:2607.14455*, 2026a.

Zhilin He, Yorai Shaoul, and Jiaoyang Li. Model-based diffusion optimal control for multi-robot motion planning. *arXiv preprint arXiv:2607.12423*, 2026b.

Neville Hogan. Impedance control: An approach to manipulation. In *1984 American control conference*, pp. 304–313. IEEE, 1984.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Wonsuhk Jung, Utkarsh A. Mishra, Nadun Ranawaka Arachchige, Yongxin Chen, Danfei Xu, and Shreyas Kousik. Joint model-based model-free diffusion for planning with constraints. In *Proceedings of the Conference on Robot Learning*, 2025.

Changliu Liu, Chung-Yen Lin, and Masayoshi Tomizuka. The convex feasible set algorithm for real time optimization in motion planning. *SIAM Journal on Control and optimization*, 56(4): 2712–2733, 2018.

Puze Liu, Davide Tateo, Haitham Bou Ammar, and Jan Peters. Robot reinforcement learning on the constraint manifold. In *Conference on Robot Learning*, pp. 1357–1366. PMLR, 2022.

Puze Liu, Haitham Bou-Ammar, Jan Peters, and Davide Tateo. Safe reinforcement learning on the constraint manifold: Theory and applications. *IEEE Transactions on Robotics*, 41:3442–3461, 2025.

Roberto Martín-Martín, Michelle A Lee, Rachel Gardner, Silvio Savarese, Jeannette Bohg, and Animesh Garg. Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 1010–1017. IEEE, 2019.

Chaoyi Pan, Zeji Yi, Guanya Shi, and Guannan Qu. Model-based diffusion for trajectory optimization. *Advances in Neural Information Processing Systems*, 37:57914–57943, 2024.

Roger Penrose. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413, 1955. doi: 10.1017/S0305004100030401.

Ralf Römer, Alexander von Rohr, and Angela P Schoellig. Diffusion predictive control with constraints. *arXiv preprint arXiv:2412.09342*, 2024.

Joohwan Seo, Nikhil PS Prakash, Xiang Zhang, Changhao Wang, Jongeun Choi, Masayoshi Tomizuka, and Roberto Horowitz. Contact-rich se (3)-equivariant robot manipulation task learning via geometric impedance control. *IEEE Robotics and Automation Letters*, 9(2):1508–1515, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6 (3):4915–4922, 2021.

Toshiaki Tsuji, Yasuhiro Kato, Gokhan Solak, Heng Zhang, Tadej Petrič, Francesco Nori, and Arash Ajoudani. A survey on imitation learning for contact-rich tasks in robotics. *The International Journal of Robotics Research*, pp. 02783649261417694, 2025.

Karl Van Wyk, Mandy Xie, Anqi Li, Muhammad Asif Rana, Buck Babich, Bryan Peele, Qian Wan, Iretiayo Akinola, Balakumar Sundaralingam, Dieter Fox, et al. Geometric fabrics: Generalizing classical mechanics to capture the physics of behavior. *IEEE Robotics and Automation Letters*, 7 (2):3202–3209, 2022.

Joe Watson and Jan Peters. Inferring smooth control: Monte carlo posterior policy iteration with gaussian processes. In *Conference on Robot Learning*, pp. 67–79. PMLR, 2023.

Joe Watson, Hany Abdulsamad, and Jan Peters. Stochastic optimal control as approximate input inference. In *Conference on Robot Learning*, pp. 697–716. PMLR, 2020.

Grady Williams, Paul Drews, Brian Goldfain, James M. Rehg, and Evangelos A. Theodorou. Information-theoretic model predictive control: Theory and applications to autonomous driving. *IEEE Transactions on Robotics*, 34(6):1603–1622, 2018.

Haoru Xue, Chaoyi Pan, Zeji Yi, Guannan Qu, and Guanya Shi. Full-order sampling-based mpc for torque-level locomotion control via diffusion-style annealing. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4974–4981. IEEE, 2025.

Lei Ye, Haibo Gao, Peng Xu, Zhelin Zhang, Junqi Shan, Ao Zhang, Wei Zhang, Ruyi Zhou, Zongquan Deng, and Liang Ding. Pegasusflow: Parallel rolling-denoising score sampling for robot diffusion planner flow matching, 2025.

Heng Zhang, Gokhan Solak, Gustavo JG Lahr, and Arash Ajoudani. SRL-VIC: A variable stiffness-based safe reinforcement learning for contact-rich robotic tasks. *IEEE Robotics and Automation Letters*, 9(6):5631–5638, 2024.

Heng Zhang, Rui Dai, Gokhan Solak, Pokuang Zhou, Yu She, and Arash Ajoudani. Safe learning for contact-rich robot tasks: A survey from classical learning-based methods to safe foundation models. *arXiv preprint arXiv:2512.11908*, 2025a.

Jichen Zhang, Liqun Zhao, Antonis Papachristodoulou, and Jack Umenberger. Constrained diffusers for safe planning and control. *arXiv preprint arXiv:2506.12544*, 2025b.

Weiye Zhao, Tairan He, and Changliu Liu. Model-free safe control for zero-violation reinforcement learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 784–793. PMLR, 2022.

Yinan Zheng, Jianxiong Li, Dongjie Yu, Yujie Yang, Shengbo Li, Xianyuan Zhan, and Jingjing Liu. Safe offline reinforcement learning with feasibility-guided diffusion model. In *International Conference on Learning Representations*, volume 2024, pp. 3838–3867, 2024.

Appendix Contents

A Related Works . . . 14
B Theoretical Proofs . . . 15
B.1 Assumptions . . . 15
B.2 Model-Based Reweighting . . . 16
B.3 Properties of Realization-Aware Refinement . . . 16
B.4 Local Residual Control . . . 17
C Environmental Setup . . . 17
C.1 Simulation Environments . . . 18
C.2 Real-World Deployment Environments . . . 19
D Baseline Details . . . 19
E Real-World Experimental Protocol and Additional Results . . . 21
E.1 Deployment Protocol . . . 21
E.2 Surface-Scanning Results . . . 21
E.3 Hole-Alignment and Insertion Results . . . 22
F Task-Specific Realization Maps . . . 22
G Additional Results . . . 23
G.1 Surface Scanning . . . 24
G.2 Peg Insertion . . . 25
G.3 Humanoid Push . . . 27
H Proposal and Refinement Details . . . 28
I Hyperparameters . . . 30
I.1 Learned Proposal Settings . . . 30
I.2 Model-Based Annealing Settings . . . 31
I.3 Realization-Aware Refinement Settings . . . 31
J Visualizations . . . 33
J.1 Surface Scanning . . . 33
J.2 Peg Insertion . . . 36
J.3 Humanoid Push . . . 36

A Related Work

Safe Reinforcement Learning and Contact-Rich Interaction. Safe RL is commonly formulated as a Constrained Markov Decision Process (CMDP), with trust-region, Lagrangian, or Lyapunov updates limiting expected cumulative costs (Achiam et al., 2017; Chow et al., 2019). Because expectation-level constraints may still permit transient violations, safe-exploration methods modify actions through learned safety layers, recovery policies, or implicit safe sets (Dalal et al., 2018; Thananjeyan et al., 2021; Zhao et al., 2022). D-ATACOM further learns uncertain long-term safety constraints within constraint-manifold exploration (Günster et al., 2024), whereas FISOR extracts hard-safe offline policies using reachability-based feasibility and diffusion modeling (Zheng et al., 2024). For contact-rich tasks, SRL-VIC jointly learns task behavior and variable stiffness with safety-critic and recovery mechanisms (Zhang et al., 2024). Together, these methods demonstrate that learned policies contain valuable task- and safety-aware interaction structure. However, they generally execute or safeguard policies at the action level. MGA instead lifts learned behavior into sequence-level proposals whose relevance is reassessed under current contact dynamics.

Model-Based Generative Models for Trajectory Sampling. Diffusion planners represent multimodal action or trajectory distributions for offline planning and imitation (Janner et al., 2022; Chi et al., 2025). Constrained variants impose test-time feasibility through denoising projections, primal–dual guidance, constraint tightening, or augmented Langevin splitting (Römer et al., 2024; Christopher et al., 2024; Zhang et al., 2025b; Blanke et al., 2025). Model-Based Diffusion (MBD) instead derives diffusion-style updates directly from objective-weighted dynamics rollouts, enabling training-free trajectory optimization and optional data guidance (Pan et al., 2024). MDOC and MD-COAS augment this process with barriers, CBF or CFS corrections, and adaptive scheduling (He et al., 2026b;a), while DIAL-MPC and PegasusFlow extend sampling-based refinement to short-horizon online optimization (Xue et al., 2025; Ye et al., 2025). Closest in spirit, JM2D aligns model-free diffusion with model-based modules through interaction-potential joint sampling (Jung et al., 2025). These methods establish learned, model-based, and hybrid generative search, but do not use an RL policy as structured proposal source for coupled motion–impedance sequences or refine such candidates through realization-aware geometry.

Constraint Geometry and Manifold-Based Optimization. Robot motion generation has long exploited non-Euclidean structure. RMPflow combines local Riemannian motion policies across task spaces, while geometric fabrics shape stable behavior through configurable metrics (Cheng et al., 2021; Van Wyk et al., 2022). Constraint-manifold methods such as ATACOM construct safe actions in tangent spaces (Liu et al., 2022; 2025), whereas Riemannian score models and flow matching define generative processes intrinsically on manifolds (De Bortoli et al., 2022; Chen & Lipman, 2024). For contact-rich manipulation, geometric impedance learning uses $\mathrm{SE}(3)$-consistent errors and gain scheduling (Seo et al., 2023), and log-Euclidean geometry provides unconstrained coordinates for positive-definite stiffness matrices (Arsigny et al., 2006). CBF-based soft-robot methods map modeled force–deformation limits into invariant pose sets (Dickson et al., 2025), while 2GO embeds active-constraint metrics, tangent stochastic filtering, and CFS retraction into model-based generative optimization (Liu et al., 2018). However, these geometries arise from prescribed constraints, task spaces, or known contact models. MGA uses feedback-conditioned task residuals to shape rollout-weighted sequence updates and correct the residuals left afterward, with response reliability modulating refinement where enabled.

B Theoretical Proofs

We provide the detailed proofs for Section 4. MBD derives a model-based clean-moment estimator and establishes small-temperature concentration of the ideal Gibbs target under sublevel-volume and minimizer separation assumptions (Pan et al., 2024, Section 4.1 and Appendix A.2). DIAL similarly relates MPPI to a Gaussian-smoothed score, followed by a Monte Carlo approximation (Xue et al., 2025, Section III-A). These results do not establish convergence of our finite, structured proposal bank. Likewise, the residual recursion in MD-COAS assumes bounds on proposal-induced residual growth and local correction error (He et al., 2026a). Below, we instead derive the finite-bank weighting identity and the residual response of the specific normalized update used by MGA.

B.1 Assumptions

A1 (Evaluated proposal bank). We condition on the $N_k \geq 1$ indexed candidates and their finite costs $j_m = \widetilde{J}(\widetilde{U}_{k,m})$. The resulting temperature $\beta_k > 0$ is fixed, including when it is computed from these costs. Repeated candidates, reused learned horizons, and deterministic anchors are allowed. Each evaluated slot retains its multiplicity.

A2 (Fixed geometric core). Within one reverse step, A_k and the symmetric positive-definite metric G_k are fixed, and $\lambda_k > 0$. No full-rank assumption on A_k is required.

A3 (Local correction chart). For Theorem 1, the feedback-conditioned chart is affine, $C^{\mathrm{real}}(U) = \mathsf{J}U - b$, on one smooth branch containing $\check{U}_k$, $V_k = \check{U}_k + \mathsf{M}_k d_k$, and V_k^+. The points lie away from clipping-switch boundaries, and the correction uses $0 < \eta \leq 1$ and $\varepsilon > 0$. Rank deficiency and an incompatible target b are allowed. The smooth nonlinear extension is stated separately below.

B.2 Model-Based Reweighting

Proof of Lemma 1. Write $N = N_k$, $\beta = \beta_k$, $u = \mathbf{1}/N$, and $Z = N^{-1}\sum_m \exp(-j_m/\beta)$. The softmax weights satisfy $\log(\bar{w}_{k,m}/u_m) = -j_m/\beta - \log Z$. Hence, for every $w \in \Delta_N$, with the convention $0 \log 0 = 0$,

$$\underbrace{\sum_m w_m j_m + \beta D_{\mathrm{KL}}(w\|u)}_{F(w)} = -\beta \log Z + \beta D_{\mathrm{KL}}(w\|\bar{w}_k). \tag{21}$$

Nonnegativity of KL divergence, with equality only at $w = \bar{w}_k$, proves the unique minimizer. Comparing $F(\bar{w}_k)$ with $F(u)$ and with a point mass on a minimum-cost candidate gives

$$\min_m j_m \le \sum_m \bar{w}_{k,m} j_m \le \min\left\{ \frac{1}{N_k}\sum_m j_m,\ \min_m j_m + \beta_k \log N_k \right\}. \tag{22}$$

The lower bound follows because the weights form a convex combination. □

This identity is the Gibbs tilt of the indexed empirical measure $N_k^{-1}\sum_m \delta_{\widetilde{U}_{k,m}}$, not an importance-corrected estimator of the continuous target in Eq. (8). No independence or proposal density is needed. In particular, the bound does not extend to $\widetilde{J}(\widehat{\bar{U}}_k)$ for a nonconvex objective.

Coverage under a fixed proposal budget. The role of the learned prior can be separated from this weighting result. Consider the initial sampling step with fixed state and proposal maps, and a measurable useful set $\mathcal{E}$ of positive thickness. Suppose independently drawn, mapped Gaussian and learned slots hit $\mathcal{E}$ with probabilities p_{g} and p_ψ, respectively. For the fixed anchor set $\mathcal{A}_k^{\mathrm{anc}}$ from Section 3.1,

$$\Pr((\mathcal{Q}_k \cup \mathcal{A}_k^{\mathrm{anc}}) \cap \mathcal{E} = \emptyset) = \mathbf{1}\{\mathcal{A}_k^{\mathrm{anc}} \cap \mathcal{E} = \emptyset\}(1-p_{\mathrm{g}})^{M_{\mathrm{g}}}(1-p_\psi)^{M_\psi}. \tag{23}$$

This follows by multiplying the independent miss probabilities. At fixed $M_{\mathrm{g}} + M_\psi = M$, replacing Gaussian slots improves or preserves coverage if $p_\psi \ge p_{\mathrm{g}}$. This is a conditional advantage, not a guarantee for any pretrained policy. The comparison concerns the initial draw with fixed maps. Cached learned horizons are not independent fresh draws at later reverse steps. Coverage alone does not order the final refined sequences.

B.3 Properties of Realization-Aware Refinement

Proof of Lemma 2. Write $G = G_k$, $A = A_k$, and $\lambda = \lambda_k$. The quadratic objective is strongly convex, and its first-order condition is $G(v-d) + \lambda^{-1}A^\top A v = 0$. The matrix inversion lemma gives

$$v = (G + \lambda^{-1}A^\top A)^{-1} G d = \left[I - G^{-1}A^\top(\lambda I + AG^{-1}A^\top)^{-1}A\right] d = \mathsf{M}_k d. \tag{24}$$

Multiplication by A yields the second identity in Eq. (19). For $B = AG^{-1/2}$,

$$G^{1/2}\mathsf{M}_k G^{-1/2} = I - B^\top(\lambda I + BB^\top)^{-1}B = (I + \lambda^{-1}B^\top B)^{-1}. \tag{25}$$

Its eigenvalues are one on $\ker B$ and $\lambda/(\lambda+\sigma_i^2) \in (0,1)$ on the right singular directions with $\sigma_i > 0$. Thus $\|v\|_G \le \|d\|_G$, and the G-orthogonal component in $\ker A$ is unchanged. Since $Ad \in \mathrm{range}(A) = \mathrm{range}(AG^{-1}A^\top)$, the response identity strictly contracts Ad whenever $Ad \neq 0$. When A is the complete active Jacobian at a regular feasible point, the limit $\lambda \downarrow 0$ recovers the metric tangent map in Eq. (4). □

Clean-moment transport. Substituting Eq. (10) into Eq. (16), and using $\bar{\alpha}_k = \alpha_k \bar{\alpha}_{k-1}$, gives

$$U_{k-1}^{\mathrm{raw}} = \frac{1}{\sqrt{\alpha_k}}\left[U_k - U_k + \sqrt{\bar{\alpha}_k}\bar{U}_{\mathcal{M},k}\right] = \sqrt{\bar{\alpha}_{k-1}}\bar{U}_{\mathcal{M},k}.$$

Consequently, the transported clean sequence is $V_k = \check{U}_k + \mathsf{M}_k d_k$, and Lemma 2 gives

$$A_k(V_k - \check{U}_k) = \lambda_k(\lambda_k I + H_k)^{-1}A_k d_k, \quad \|A_k(V_k - \check{U}_k)\|_2 \le \|A_k d_k\|_2. \tag{26}$$

The inequality is strict when $A_k d_k \neq 0$. This identity concerns the chosen deterministic transport. Arbitrary reverse transports or additional reverse noise need not preserve this samplewise bound.

From encoded responses to chart responses. Lemma 2 controls $A_k d$, not necessarily the full Jacobian response. To quantify this distinction, write $A = A_k$, $G = G_k$, $\lambda = \lambda_k$, and let M_A, M_J be the same regularized filter constructed from A and a fixed chart Jacobian J, respectively. Set $B = AG^{-1/2}$, $D = \mathsf{J}G^{-1/2}$, and $\delta = \|B^\top B - D^\top D\|_2$. With $R_B = (I + B^\top B/\lambda)^{-1}$ and $R_D = (I + D^\top D/\lambda)^{-1}$, the resolvent identity gives $R_B - R_D = \lambda^{-1} R_B (D^\top D - B^\top B) R_D$. Both resolvents have norm at most one, so $\|(\mathsf{M}_A - \mathsf{M}_\mathsf{J})d\|_G \le (\delta/\lambda)\|d\|_G$. If D has positive rank, Lemma 2 applied to J and the triangle inequality yield

$$\|\mathsf{J}\mathsf{M}_A d\|_2 \le \frac{\lambda}{\lambda + \sigma^2_{\min,+}(D)} \|\mathsf{J}d\|_2 + \frac{\|D\|_2 \delta}{\lambda} \|d\|_G. \tag{27}$$

The zero-rank case is trivial. This bound makes the required encoding accuracy explicit. It does not assume that the implemented dictionary has small δ. Its construction and limitations are described in Appendix H.

B.4 Local Residual Control

Proof of Theorem 1. On the fixed affine branch, abbreviate $C = C^{\text{real}}$ and define $T_\varepsilon := I - \eta \mathsf{J}\mathsf{J}^\top (\mathsf{J}\mathsf{J}^\top + \varepsilon I)^{-1}$. Substitution of Eq. (17) gives the exact identity

$$C(V_k^+) = T_\varepsilon C(V_k) = T_\varepsilon \left[C(\check{U}_k) + \mathsf{J}\mathsf{M}_k d_k \right]. \tag{28}$$

The orthogonal projector $\Pi = \mathsf{J}\mathsf{J}^\dagger$ commutes with T_ε. On each left singular direction of J with singular value $\sigma_i > 0$, the latter has eigenvalue $1 - \eta\sigma_i^2/(\sigma_i^2 + \varepsilon) \in [0, \rho]$. Therefore $\|\Pi C(V_k^+)\|_2 \le \rho \|\Pi C(V_k)\|_2$, and $\Pi\mathsf{J} = \mathsf{J}$ proves Eq. (20). On the orthogonal complement, T_ε is the identity and $(I - \Pi)C(U) = -(I - \Pi)b =: c_\perp$ is constant. Equivalently,

$$\|C(V_k^+)\|_2^2 - \|c_\perp\|_2^2 \le \rho^2 \left[\|C(V_k)\|_2^2 - \|c_\perp\|_2^2 \right]. \tag{29}$$

If $\mathsf{J} = 0$, the correction is zero and $C(V_k^+) = C(V_k) = -b$. Thus redundant rows require no full-row-rank assumption, while incompatible components cannot be removed by this correction. □

The task charts in Appendix F are affine after fixing the replanning context and the clipping branches. Theorem 1 therefore applies when the update remains on one such branch. It compares V_k^+ with V_k, not with $\check{U}_k$: the proposal displacement may introduce residual before correction. Neither branch switching nor a sequence of changing proposal banks is covered by a fixed-chart convergence claim.

Smooth nonlinear charts. Let $C = C^{\text{real}}$ have an L-Lipschitz Jacobian on the segment from V_k to $V_k^+ = V_k + p$, where p is the correction in Eq. (17), with $0 < \eta \le 1$ and $\varepsilon > 0$. With $\mathsf{J} = \mathsf{J}(V_k)$, the integral Taylor formula gives

$$C(V_k^+) = T_\varepsilon C(V_k) + e, \quad e = \int_0^1 [\mathsf{J}(V_k + sp) - \mathsf{J}(V_k)] p \, ds, \quad \|e\|_2 \le \frac{L}{2}\|p\|_2^2.$$

For $\Pi = \mathsf{J}\mathsf{J}^\dagger$ and the positive-rank factor ρ defined in Theorem 1, the same orthogonal decomposition yields

$$\|C(V_k^+)\|_2 \le \sqrt{\|(I - \Pi)C(V_k)\|_2^2 + \rho^2 \|\Pi C(V_k)\|_2^2} + \frac{L}{2}\|p\|_2^2. \tag{30}$$

Thus the linearized reduction persists when it exceeds the second-order remainder. For example, if $C(V_k) \in \text{range}(\mathsf{J})$ and $L\|p\|_2^2/2 < (1 - \rho)\|C(V_k)\|_2$, the full residual strictly decreases. Unlike the affine case, $(I - \Pi)C(V_k)$ is only a local linearization component, not a fixed residual floor for the nonlinear chart. If $\mathsf{J}(V_k) = 0$, then $p = 0$ and the correction leaves V_k unchanged.

C Environmental Setup

We provide detailed descriptions of the environments used in this work. The simulation benchmarks isolate three aspects of contact-rich control: **(1)** adaptation of learned insertion behavior to model and sensing mismatch, **(2)** coordination of path progress, force, and deformation under changing surface response, and **(3)** whole-body balance during object interaction. The real-world environments examine deployment of the model-based refinement component under measured contact feedback.

C.1 Simulation Environments

PegInsert. A Franka Panda inserts a rectangular peg into a fixed socket by coordinating pose alignment, impedance, and axial force. The peg has a 20×12-mm cross-section and a 50-mm length. The socket is 40 mm deep, with 1.5-mm nominal clearance on each side. The peg starts 4 mm above the socket. We use a 10-N axial-force reference, force limits of 30 N axially and 20 N laterally, and bending/torsional torque limits of $1.5/1.0$ N m. The three evaluation conditions retain the same task objective but change the source of mismatch:

- **ID:** We use nominal socket geometry and unbiased wrench observations.
- **Pose-OOD:** Execution-side socket translations are perturbed independently by up to 1 mm in each lateral direction, and orientation components by up to $0.7°$, while the planner retains the nominal model. Clearance and friction are unchanged.
- **Sensing-OOD:** Execution observations include force bias $(0.5, -0.5, 0)$ N and torque bias $(0.02, -0.02, 0.01)$ N m, without changing the physical task geometry.

Completion requires at least 32 mm of insertion, lateral error at most 1.2 mm, orientation error at most 0.06 rad, and instantaneous wrench compliance for three consecutive control steps. Strict safe success further requires a complete 64-step execution with no wrench violation or jam anywhere in that window, including steps after task completion. Execution continues through all 64 steps. Incomplete or aborted runs count as failures. Safety metrics use the true simulated wrench rather than the biased observations. This distinction separates successful insertion from unsafe progress and isolates the role of model-based reassessment of learned behavior.

Surface Scanning. A Panda follows a prescribed surface path while regulating normal force and limiting deformation. We consider five shape families—plane, cylinder, convex, bumpy, and held-out randomly generated surfaces—under both rigid and compliant contact. Three additional planar environments vary material response spatially through alternating hard–soft stripes, a hard center, or a soft center, giving 13 conditions in total. Force references are 20 N for rigid and hybrid surfaces, 18 N for the four nominal compliant shapes, and 15 N for the unseen compliant surface. The force limit is 60 N. The unseen setting also varies friction and compliance.

The main table partitions these conditions into four disjoint groups. **Hard** contains rigid plane, cylinder, convex, and bumpy surfaces. **Soft** contains their four compliant counterparts. **Hybrid** contains the three spatial stiffness maps. **Unseen** contains the held-out rigid and compliant surfaces. Unseen trials are not also counted in Hard or Soft.

Rigid and compliant surfaces use MuJoCo probe–surface contact. The hybrid environments instead use a spatial elastic-foundation response with hard and soft stiffnesses of 8000 and 2000 N/m, respectively, without an additional probe collision response. This separates material transitions from changes in surface shape. We evaluate realized in-contact path progress, force tracking, contact loss, and deformation, rather than commanded path progress alone. These environments test whether refinement follows a physically responsive direction when similar commands produce different contact motions.

Scanning retains the operational SSR criterion frozen after the two-seed layout study: contact-valid trajectory coverage must be at least 90%, settled contact loss at most 10%, and force-violating samples at most 1%. Coverage uses the complete executed path with a 5-mm tolerance. Only contact-loss measurement excludes the first ten acquisition steps. The force tolerance allows at most one of the 100 control-step samples to exceed 60 N and is not a zero-violation criterion. Tail utilization is the mean of the five largest actual normal-force samples divided by 60 N. It does not measure deformation or force-tracking error.

Humanoid Push. A Unitree H1 coordinates hand contact, whole-body balance, and box motion. The reported comparison contains three conditions. **Force Regulation** tracks a 15-N hand-force reference against an effectively stationary box, isolating force control from object displacement. **Fixed-stance Push** moves a nominal 8-kg box 0.10 m from a fixed stance. **Unjamming** releases an initially rotated box from a narrow corridor and advances it 0.03 m, with a final yaw tolerance of 0.03 rad. The 30-N force step, randomized pushing conditions, and walking pushes are not part of this comparison.

Force-regulation SSR requires reaching 90% of the force reference, staying within 15 ± 1.5 N over the terminal 0.20 s, and avoiding physical-safety violations. This criterion measures force-regulation success rather than box displacement. Fixed-stance pushing and unjamming require task completion without physical-safety violations, including excessive hand force, unintended body–object contact, or violations of balance and fall limits. Safety checks include the initial state and all active 4-ms physics samples. The normalized tail metric uses physical hand force divided by the 60-N limit and averages the largest $\lceil 0.05N \rceil$ samples. Post-success padding is excluded. It does not subsume the separate balance and contact constraints.

C.2 Real-World Deployment Environments

The hardware platform is a 6-DoF Dobot Nova 5 with a wrist-mounted 6-axis force/torque sensor and a probe surrogate having a 40-mm-diameter contact head and a 200-mm nominal length. We use three scanning surfaces and one insertion fixture to separate material compliance, curved-surface following, and pre-contact geometric alignment (Fig. 6).

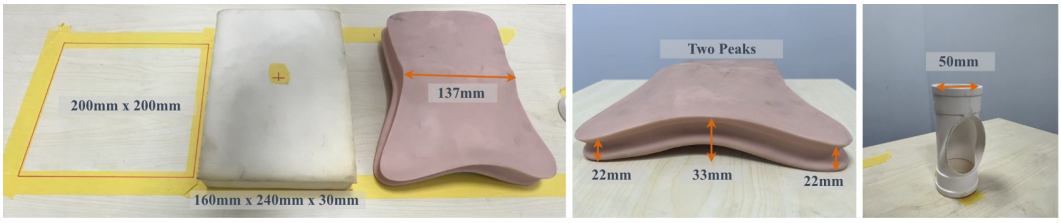


Figure 6: Real-world contact fixtures: the rigid table region, compliant foam, and curved phantom (left), the phantom profile (center), and the pipe used for clearance insertion (right). Dimensions describe the fixtures rather than the commanded path lengths.

Table. A rigid planar table provides a reference condition for contact establishment and steady force tracking. The robot traverses a 100-mm path with a 3-N normal-force reference.

Foam. A nearly planar compliant foam block uses the same 100-mm path with a lower 0.8-N force reference. This environment emphasizes contact recovery and the tradeoff between tangential progress and force regulation.

Phantom. A curved anatomical phantom requires a 110-mm scan at a 3-N force reference. The controller estimates the local normal from recent contact positions and adjusts tool orientation, without a pre-reconstructed surface mesh or live camera feedback.

Hole Insertion. A 50-mm-bore pipe accommodates the 40-mm tool envelope, giving a nominal 5-mm radial clearance. The robot begins 100 mm above the rim with an approximately 40-mm lateral offset, aligns with the pipe axis, and then inserts the tool. This is a clearance-insertion task, distinct from the tight-clearance simulated PegInsert benchmark.

These are bench contact-control experiments with a surrogate tool, not ultrasound-image acquisition or human-subject experiments. The deployment protocol and quantitative results are reported in Appendix E.

D Baseline Details

We compare action-level safety correction and sampling-based trajectory optimization to separate local constraint handling from model-based horizon refinement. Table 2 summarizes the task-adapted implementations and the MGA variants. The descriptions below explain how each method uses the common contact-control interface. The entries do not imply global safety guarantees.

ISSA. The implicit safe-set algorithm (Zhao et al., 2022) filters the same nominal PPO action through a derivative-free search. Its safety test queries one-step transitions to discourage unsafe set crossings or avoid increasing the safety index when already outside the safe set. It does not optimize

	Method	Policy Role	Model-based	Horizon Rollout	Constraint Treatment	Realization Geometry	LRC	Learned Reliability
Baselines	ISSA	Nominal action	✗	✗	Implicit safe-set filter	✗	✗	✗
	ATACOM	Tangent action	✗	✗	Augmented constraint manifold	✗	✗	✗
	MPPI	✗	✓	✓	Reward penalties	✗	✗	✗
	DIAL	✗	✓	✓	Reward penalties	✗	✗	✗
	PegasusFlow	✗	✓	✓	Reward penalties	✗	✗	✗
Variants	**MGA**	Structured proposals	✓	✓	Feasibility cost + local correction	✓	✓	✓
	MGA w/o RL prior	✗	✓	✓	Feasibility cost + local correction	✓	✓	✓
	MGA w/o LRC	Structured proposals	✓	✓	Feasibility cost without local correction	✓	✗	✓

Table 2: The table compares the configured simulation baselines and MGA variants. Entries describe the task-adapted implementations, with ✓ and ✗ denoting presence and absence of the indicated mechanism. Model-based denotes use of an explicit dynamics model to predict and optimize candidate trajectories. Black-box safety queries and local constraint geometry do not qualify. Realization geometry and LRC denote response-conditioned sequence refinement and its post-transport residual correction, not action-level constraint handling.

a trajectory horizon. Thus, ISSA tests action-level correction rather than joint proposal sampling and geometric sequence refinement.

ATACOM. ATACOM (Liu et al., 2022) transforms a learned free action into a direction on an augmented constraint manifold, using slack variables for inequalities and a local correction for manifold drift. We adapt this construction to the common contact-control primitive and train a dedicated PPO policy through the tangent-action interface. It is not a post-hoc projection of the ordinary PPO policy. This baseline distinguishes action-level constraint geometry from feedback-conditioned sequence refinement.

MPPI. Model Predictive Path Integral control (Williams et al., 2018) perturbs dense control sequences with Gaussian noise and combines them using rollout-return weights. The task reward penalizes undesirable contact behavior, but the update has neither a learned proposal prior nor the realization-aware geometric filter.

DIAL. DIAL (Xue et al., 2025) uses diffusion-style annealing of horizon-node perturbations with reward-weighted sequence updates. We retain this sampling mechanism under the common task interface, without MGA's learned proposals, additional feasibility cost, or local manifold correction. It tests whether annealing alone suffices under contact-response mismatch.

PegasusFlow. PegasusFlow (Ye et al., 2025) performs rolling denoising with adaptive basis-weighted updates of a spline-parameterized horizon. Like DIAL, it evaluates candidate controls through task rollouts but does not use the learned proposal or realization-aware correction of MGA.

No RL Prior (w/o RL prior). This ablation removes the proposal policy while retaining the remaining refinement and execution-selection mechanisms of MGA.

No Local Residual Correction (w/o LRC). This ablation disables the correction in Eq. (17) while retaining learned proposals, model-based rollout weighting, geometric shaping, realization-aware geometry, and execution acceptance.

MGA. Our method uses the policy to supply structured candidate horizons rather than committing to its action. Under replacement, rollout weighting combines learned and Gaussian candidates before realization-aware refinement. Under additive integration, separately evaluated learned horizons compete with the refined Gaussian sequence through execution acceptance. Comparing with the no-prior and no-LRC ablations therefore tests the complementary roles of learned structure, model-based reassessment, and local correction.

Common comparison settings. Methods share the task interface, control period, evaluation conditions, and random seeds. Sampling-based methods use a nominal budget of 64 stochastic proposals and the same refinement-iteration settings within each benchmark. Method-specific anchors, safety queries, and revalidation are retained. This matches the nominal sampling budget, not the total computational cost. In all three PegInsert suites and the reported humanoid runs, the prior is evaluated additively: one deterministic and eight stochastic policy proposals are evaluated once per replanning call after Gaussian refinement and revalidation, with 64 Gaussian samples per refinement iteration. The selected expert can replace the refined Gaussian sequence if accepted. These expert horizons do not enter the Gaussian refinement bank or its geometric correction. Its comparison with the no-prior ablation therefore does not establish an advantage at equal total rollout cost. The environment reward is shared, while additional feasibility weighting is part of the MGA variants. In PegInsert, ISSA and MGA use the same frozen PPO policy. ATACOM uses a separately trained tangent-space policy with the same two-layer, 16-unit architecture. Both policies use $200{,}000$ environment steps of training and select the final-budget checkpoint after an independent development health gate. Each policy's checkpoint hash is locked across all three PegInsert suites. No policy is updated on the evaluation trials.

E Real-World Experimental Protocol and Additional Results

We report the deployment protocol and additional measurements for the environments in Appendix C.2. These experiments evaluate the model-based refinement component of MGA with contact feedback. The hardware configuration does not use a learned proposal prior.

E.1 Deployment Protocol

The planner uses a 0.4-s horizon, 48 candidate sequences, and three initial or two subsequent refinement iterations. Planning and position-command execution have nominal rates of 10 and 20 Hz, respectively. A bounded normal-force feedback controller converts the selected force reference into position corrections, while tangential progress depends on the measured contact response. This is a position-based realization of the MGA deployment component, not direct execution of a full Cartesian stiffness matrix. The observed behavior reflects the planner and feedback controller together.

For surface scanning, each trial begins with unloaded force/torque zeroing and a slow approach to the taught path. After contact is established and force settles near its reference, the robot traverses the path at a nominal commanded speed of 5 mm/s. Persistent force-tracking error pauses tangential progress while bounded normal recovery continues. The robot releases contact after reaching the endpoint. Table and foam retain the initial tool orientation, whereas the phantom updates orientation using the estimated local surface normal. Recovery pauses are part of the task and are included in the elapsed scanning time.

Forces are gravity compensated and tared before each trial. The calibration has a held-out 95th-percentile force residual of 1.14 N. Reported tracking errors therefore characterize the compensated feedback signal rather than independently verified absolute contact force, particularly for the low-force foam condition.

E.2 Surface-Scanning Results

We retain all five recorded trials for each surface. All 15 scans reach their path endpoint. Complete-session outcomes, including release and final positioning, are $5/5$ for table, $4/5$ for foam, and $5/5$ for phantom. The remaining foam session stops because robot feedback becomes stale after scanning and release. Its completed scan remains included in the force statistics.

The evaluation interval begins with tangential scanning and ends at scan completion, including intermediate holds and contact recovery but excluding initial approach, force settling, and final release. For compensated normal force samples $F_{n,i}$, reference $F_n^\star$, and actual sampling intervals Δt_i, we compute

$$\overline{F}_n = \frac{\sum_i \Delta t_i F_{n,i}}{\sum_i \Delta t_i}, \quad \mathrm{MAE} = \frac{\sum_i \Delta t_i |F_{n,i} - F_n^\star|}{\sum_i \Delta t_i}, \quad \mathrm{RMSE} = \sqrt{\frac{\sum_i \Delta t_i (F_{n,i} - F_n^\star)^2}{\sum_i \Delta t_i}}. \tag{31}$$

Table 3 reports the mean and sample standard deviation of these per-trial quantities over five repetitions. The main-text figure reports the same across-trial mean-force statistics.

Surface	$F_n^\star$ (N)	Mean F_n (N)	MAE (N)	RMSE (N)
Table	3.0	2.987 ± 0.014	0.391 ± 0.054	0.506 ± 0.094
Foam	0.8	0.817 ± 0.017	0.176 ± 0.006	0.222 ± 0.012
Phantom	3.0	2.940 ± 0.064	0.378 ± 0.013	0.454 ± 0.015

Table 3: Surface-scanning results report the mean $\pm$ sample standard deviation over five trials per material. Metrics use unsmoothed compensated feedback and physical timestamps, including recovery intervals. Complete-session outcomes are reported separately in the text.

Mean force remains close to the reference on all three surfaces, but the lower absolute error on foam does not imply easier regulation. Its mean RMSE is approximately 27.8% of the force reference, compared with 16.9% for table and 15.1% for phantom, and its elapsed scanning time is longer. The phantom additionally requires height and orientation adaptation along a curved contact path. These results support coordinating progress with measured contact response rather than executing a fixed-height trajectory at constant speed. All reported metrics use unsmoothed signals and physical elapsed time.

E.3 Hole-Alignment and Insertion Results

The robot tracks a descending alignment reference from above the pipe rim before entering the bore. Entry requires a robot-reported lateral offset below 0.5 mm and successful orientation checks. Insertion then follows the taught axis with a 0.75-mm offset limit. Axial and lateral force soft/hard thresholds are $3/6$ N and $2/4$ N, respectively, with torque thresholds of $0.5/1.0$ N m. These bounds govern the monitored execution, not a claim of exact geometric clearance.

For pipe-axis location (x_c, y_c) and robot-reported tool/flange reference $(x(t), y(t))$, the lateral alignment measure is $e(t) = \sqrt{(x(t) - x_c)^2 + (y(t) - y_c)^2}$. In the recorded trial, pre-entry refinement reduces e from 40.31 mm to 0.011 mm at the rim. The robot then completes approximately 107 mm of insertion over 55.32 s, with maximum in-pipe compensated axial and lateral forces of 0.703 and 1.329 N. The offset is an internal robot-coordinate measure, not an external measurement of shaft-to-wall clearance or absolute mechanical accuracy.

Only one insertion trial is reported, and contact-force refinement is not activated in that trial. It therefore demonstrates pre-entry geometric alignment followed by monitored clearance insertion, rather than jam recovery or sustained force-regulated insertion. Together with the repeated surface trials, it provides deployment evidence for the model-based component of MGA. The contribution of the learned prior is evaluated separately through the controlled simulation comparisons.

F Task-Specific Realization Maps

The main text uses one residual C^{real} for all tasks. Here t_0 denotes the current control-step index and $h = 0, \ldots, T-1$ the local prediction index, corresponding to control step t_0+h. The implementation represents U by $N_u = H_{\mathrm{node}} + 1$ control nodes of dimension d_u, so $d_U = N_u d_u$. A fixed quadratic interpolation $\mathcal{I}$ produces $U^{\mathrm{dense}} = \mathcal{I}U$ with $T = H_{\mathrm{sample}} + 1$ controls. Thus the dense sequence in the preliminaries is U^{dense}, while the method and theory operate on node coordinates with the rollout objective and C^{real} composed with $\mathcal{I}$. Their derivatives include this fixed interpolation. Let $u_h = (r_h, \upsilon_h^S, \nu_h)$ denote a dense control, with $s_{\mathrm{scale}}\upsilon_h^S$ representing $S_h - S^{\mathrm{ref}}$ in symmetric coordinates.

For the two arm tasks, the current realized-minus-command tracking error e_{t_0} and local response map B_{t_0} define a frozen correction $\delta_{t_0} = -B_{t_0}^\top (B_{t_0} B_{t_0}^\top + \lambda_B I)^{-1} e_{t_0}$, where $\lambda_B > 0$. The map is estimated by central differences of terminal task coordinates under constant raw motion offsets over the first H_B controls of the incumbent. It is 2×2 for surface coordinates and 6×6 for insertion pose coordinates. The scanning error is the tangent tracking error expressed in surface coordinates with the configured along- and cross-path weights. Let $\bar{\delta}_{t_0}$ be the componentwise bounded correction multiplied by the task reliability gate, which is one when disabled. The horizon ramp is $b_h = \min\{(h+1)/H_B, 1\}$. Mapping $\bar{\delta}_{t_0}$ to task coordinates gives the target shifts below. The current state, response map, gates, and target shifts are held fixed while differentiating the chart. They change the clean chart rather than the proposal bank.

For raw force coordinate ν, the arm tasks use $F_{\mathrm{cmd}}(\nu) = F_{\mathrm{lo}} + \frac{1}{2}(\mathrm{clip}_{[-1,1]}(\nu) + 1)(F_{\mathrm{hi}} - F_{\mathrm{lo}})$, where $F_{\mathrm{lo}} = F_{\min} - F_{\mathrm{pad}}$ and $F_{\mathrm{hi}} = F_{\max} + F_{\mathrm{pad}}$ include the configured command padding. Humanoid tasks use the same map with $F_{\mathrm{pad}} = 0$, or the constant $F_{\mathrm{cmd}} = F^\star$ when the force target is fixed. Here $F^\star$ is the task's desired normal force. These maps are affine on each clipping branch.

Surface scanning. Let $q_{t_0} = (\xi_{t_0}, \eta_{t_0})$ be the current commanded surface coordinate and let $q_h(U) = \mathrm{clip}_{[0,1]^2}(q_{t_0} + a_q \sum_{\ell=0}^{h} r_{\ell,1:2})$ be its cumulative horizon chart, where a_q converts raw motion into surface-coordinate increments. The shift $\Delta^q_{t_0,h}$ is b_h times $a_q H_B \bar{\delta}_{t_0}$ after the configured correction gain and coordinate clipping. It defines $\widetilde{q}^\star_{t_0,h} = q^\star_{t_0,h} + \Delta^q_{t_0,h}$. For commanded-force map $F_{\mathrm{cmd}}(\nu_h)$ and $s_F = F_{\max} - F_{\min}$, the residual is

$$C^{\mathrm{scan}}_{t_0}(U) = \mathrm{stack}_h \left(q_h(U) - \widetilde{q}^\star_{t_0,h}, \frac{F_{\mathrm{cmd}}(\nu_h) - F^\star}{s_F} \right). \tag{32}$$

Peg insertion. Let $p^{\mathrm{cmd}}_{t_0}$ be the current six-dimensional commanded pose and $p_h(U) = p^{\mathrm{cmd}}_{t_0} + \sum_{\ell=0}^{h} D_p r_\ell$, where D_p is the diagonal map from raw translation and rotation channels to pose increments. The response-aware target is $\widetilde{p}^\star_{t_0,h} = p^\star_{t_0,h} + \Delta^p_{t_0,h}$, where $p^\star_{t_0,h}$ is the approach–insertion reference and $\Delta^p_{t_0,h} = b_h H_B D_p \bar{\delta}_{t_0}$. The chart uses diagonal pose scaling D_C with translation scales $(0.01, 0.01, d_{\mathrm{socket}})$ m and rotation scales $(0.1, 0.1, 0.1)$ rad, where d_{socket} is the socket depth. With $s_F = \max\{F_{\max}, 1\}$, its contact activation is $w_{t_0,h} = \sigma(\widetilde{p}^\star_{t_0,h,z}/s_z)$, where $\sigma(x) = (1 + e^{-x})^{-1}$ and $s_z = 0.001$ m is the activation scale:

$$C^{\mathrm{insert}}_{t_0}(U) = \mathrm{stack}_h \left(D_C^{-1} \left[p_h(U) - \widetilde{p}^\star_{t_0,h} \right], w_{t_0,h} \frac{F_{\mathrm{cmd}}(\nu_h) - F^\star}{s_F} \right). \tag{33}$$

Humanoid pushing. The whole-body task uses direct contact-semantic coordinates rather than an incremental end-effector response correction. Define $c^S_h := \mathrm{svec}(S_h - S^{\mathrm{ref}}) = s_{\mathrm{scale}} v^S_h$ and $\varphi_h := (F_{\mathrm{cmd}}(\nu_h) - F^\star)/(F_{\max} - F_{\min})$. Its chart is $C^{\mathrm{push}}_{t_0} = \mathrm{stack}_h\, C^{\mathrm{push}}_{t_0,h}$, with $\Delta^{\mathrm{real}}_{t_0} = 0$ and

$$C^{\mathrm{push}}_{t_0,h}(U) = \begin{cases} [c^S_h, \varphi_h]^\top, & \text{force regulation or fixed-stance pushing,} \\ [c^S_h, \varphi_h, (r^{\mathrm{contact}}_h)^\top]^\top, & \text{unjamming,} \\ [\varphi_h], & \text{walking push,} \end{cases} \tag{34}$$

Here svec vectorizes a symmetric matrix and r^{contact}_h denotes the face-selection logits. The remaining contact coordinates are optimized through rollout evaluation rather than fixed by this chart. Measured contact reliability is recorded separately, and the direct clean chart is unchanged. In the fixed-force suites, φ_h is identically zero. It may be omitted in the local constant-rank interpretation, although the implementation retains the row and handles its redundancy through regularization.

Soft candidate evaluation. For the soft evaluation cost in Section 2.2, let $v^h \geq 0$ collect task-selected equality magnitudes and positive inequality violations along the rollout. Following MD-COAS (He et al., 2026a), we use $\mathcal{V}(U; s^0, c) = \sum_{h=0}^{T-1} \left(\lambda^\top v^h + \frac{\rho}{2} \|v^h\|_2^2 \right)$, with $\lambda \succeq 0$ and $\rho \geq 0$. These evaluation residuals are distinct from the realization charts above.

G Additional Results

We examine realized path fidelity, productive insertion, and continued whole-body interaction. All three environments use seeds 0–9, with the evaluation rules in Appendix C.

G.1 Surface Scanning

Realized coverage beyond terminal progress. Removing LRC yields 99.88% mean terminal progress, but reduces contact-valid coverage from 96.03% to 84.75% across the 13 conditions (Fig. 7). Soft-center coverage falls from 100% to 57.43%. Terminal progress projects the final position onto the reference, whereas coverage counts reference waypoints visited with valid contact within 5 mm. Their disagreement shows why advancing the endpoint is insufficient: correction must preserve the executed path.

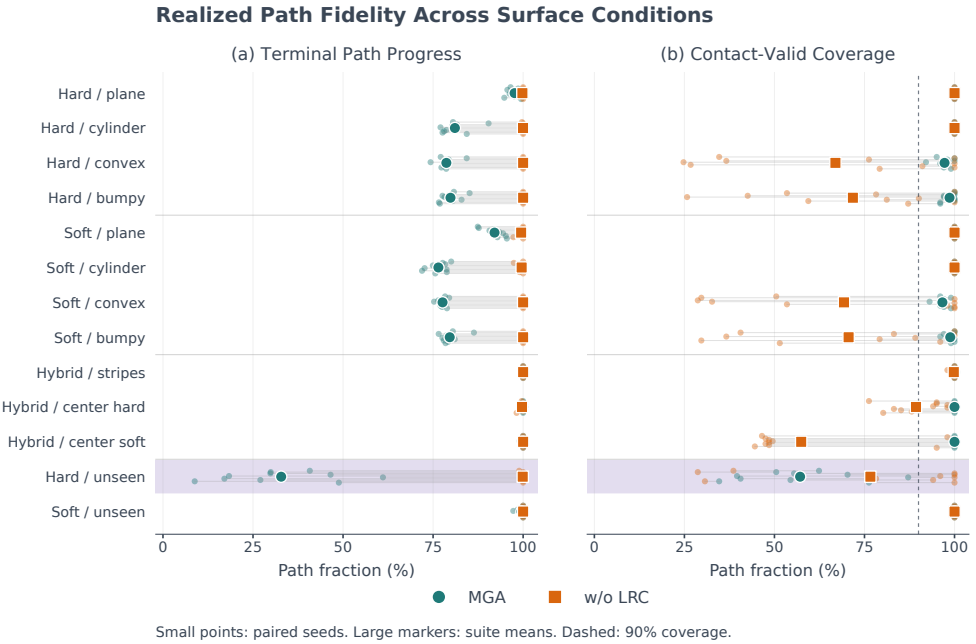


Figure 7: The plots compare terminal progress and contact-valid coverage across all scanning conditions with matched settings and seeds. Large markers show suite means, and small markers retain individual executions. Held-out surfaces are not included again in the Hard and Soft groups.

Sensitivity to the success definition. Disallowing the operational criterion's single force-violating sample leaves the coverage benefit intact (Table 4). Of the 43 no-LRC failures, 42 fail coverage alone and one fails settled contact: path fidelity, rather than the force allowance, dominates.

Method	Operational success	Zero-violation success	Coverage (%)
MGA	120/130	119/130	96.03 ± 1.38
w/o RL prior	120/130	117/130	97.10 ± 1.26
w/o LRC	87/130	86/130	84.75 ± 6.19

Table 4: Scanning success is evaluated under two force-violation allowances. Both criteria require coverage $\geq 90\%$ and settled contact loss $\leq 10\%$. Counts cover all 13×10 executions. Coverage is reported as the mean and population standard deviation of the ten equal-suite seed means.

Progress, contact load, and deformation. On Soft Convex, MGA reduces force-tail utilization from 0.432 to 0.388 and deformation CVaR_{95} from 0.548 to 0.493 mm relative to w/o LRC while recovering coverage (Fig. 8). Adding the prior also lowers Hybrid force tails from 0.738 to 0.658. These benefits are conditional: on unseen rigid contact, removing LRC improves coverage from 57.13% to 76.63%.

Spatial evidence of realized path fidelity. Figure 9 exposes the execution-level effect on a compliant curved surface. MGA visits 96.04% of the path with valid contact, versus 50.50% without LRC, although terminal progress is 78.33% and 100%, respectively. The no-LRC execution advances while leaving the tracking tolerance, rather than faithfully covering the path.

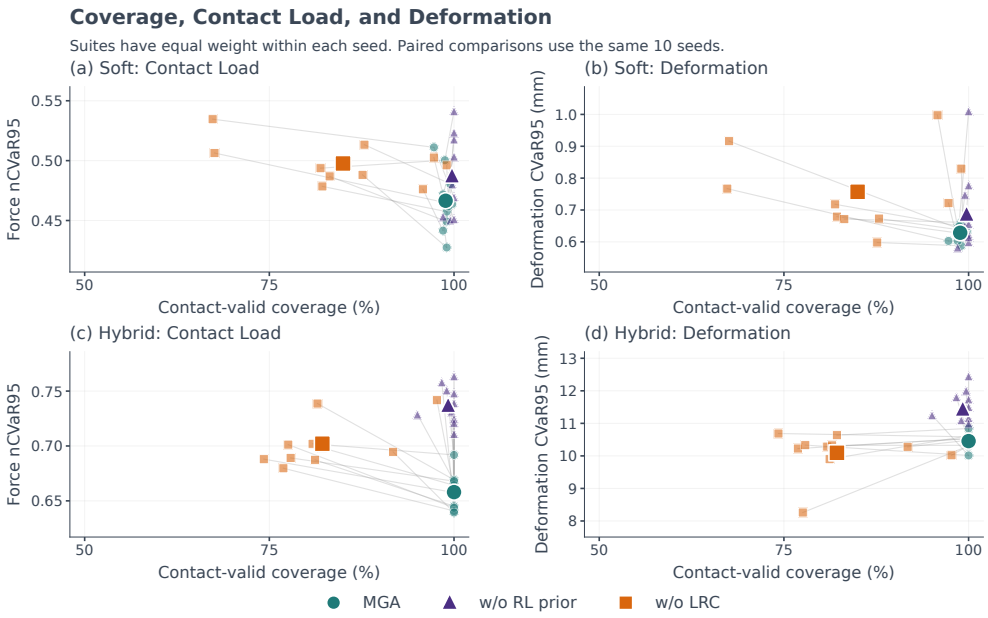


Figure 8: Matched seed-level observations compare coverage, force tails, and deformation for MGA and variants without the learned prior or LRC under compliant and hybrid contact. Force utilization is normalized by the 60-N limit, and deformation is reported in physical units.

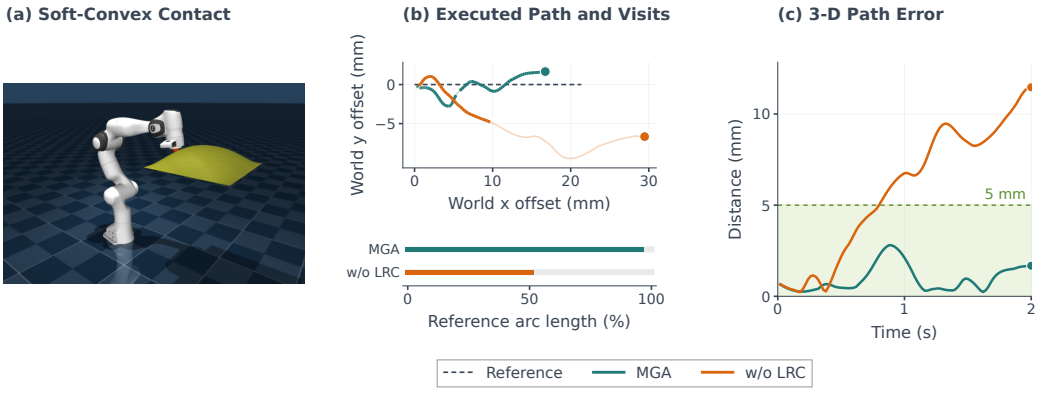


Figure 9: The Soft Convex panels show native MGA context and matched executions. Visit strips count reference waypoints within 5 mm of contact-valid executed samples in 3-D. The error trace measures executed-point distance to the reference polyline, also in 3-D. The boundary marks the same tolerance.

G.2 Peg Insertion

Completion and contact safety. Reaching the target and completing safely are separate outcomes (Fig. 10). The contact utilization is

$$\rho_t = \max\left\{\frac{\|F_{xy,t}\|}{20\,\mathrm{N}}, \frac{|F_{z,t}|}{30\,\mathrm{N}}, \frac{\|\tau_{xy,t}\|}{1.5\,\mathrm{N\,m}}, \frac{|\tau_{z,t}|}{1.0\,\mathrm{N\,m}}\right\}. \tag{35}$$

We evaluate the recorded simulated loads even under sensing bias. For each 64-step execution, nCVaR_{95} averages the four largest values of ρ_t. Safe completion requires success with no wrench violation or jam in this complete window, including after first completion. MGA completes $28/30$ trials safely, compared with 23 for PegasusFlow, 17 each for MPPI and DIAL, 5 for ATACOM, and 0 for ISSA.

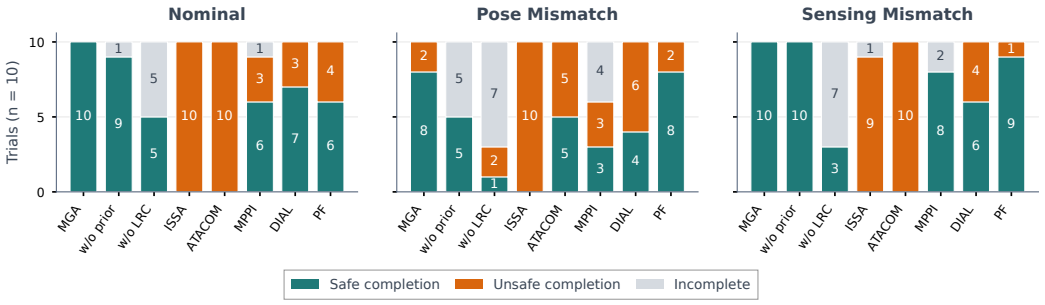


Figure 10: Insertion outcomes are reported across ten seeds per method and setting. All eight methods use the same frozen evaluation conditions. Safe completion, completion with a wrench violation or jam, and incomplete insertion are reported separately. No trial is discarded.

When learned behavior supports productive insertion. Under Pose-OOD, the prior raises raw completion from $5/10$ to $10/10$ and safe completion from $5/10$ to $8/10$, while mean tail utilization rises from 0.769 to 0.893. Under ID it raises safe completion from $9/10$ to $10/10$. Under Sensing-OOD both variants complete $10/10$ safely. Figure 11 retains incomplete trials to distinguish low-load stalling from successful execution.

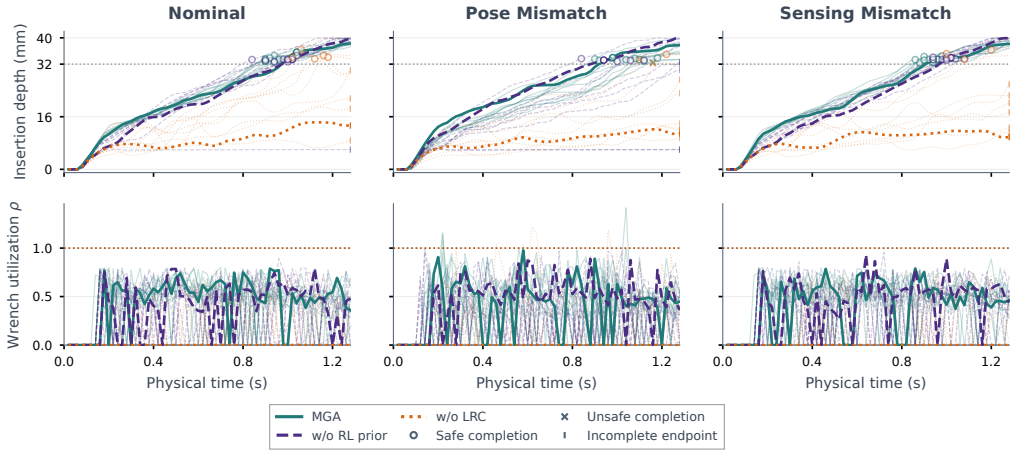


Figure 11: Depth and wrench utilization for ten seeds per variant and setting, with seed 0 emphasized. All recorded steps are retained. The depth and $\rho = 1$ thresholds do not subsume pose tolerances or the three-step hold.

Setting	Method	Completed	$F_{xy}^{\max}$ (N)	$F_{z}^{\max}$ (N)	F/T violation (%)
ID	MGA	$10/10$	15.85 ± 0.13	14.51 ± 1.18	0.00 ± 0.00
	w/o RL prior	$9/10$	15.14 ± 1.93	15.04 ± 1.12	0.00 ± 0.00
	w/o LRC	$5/10$	10.46 ± 6.14	7.98 ± 5.05	0.00 ± 0.00
Pose-OOD	MGA	$10/10$	20.05 ± 3.09	18.00 ± 3.48	0.62 ± 1.25
	w/o RL prior	$5/10$	17.17 ± 3.02	16.73 ± 2.46	0.00 ± 0.00
	w/o LRC	$3/10$	12.46 ± 9.05	10.01 ± 8.23	0.78 ± 1.88
Sensing-OOD	MGA	$10/10$	16.04 ± 0.58	14.40 ± 1.22	0.00 ± 0.00
	w/o RL prior	$10/10$	16.52 ± 1.25	15.77 ± 2.02	0.00 ± 0.00
	w/o LRC	$3/10$	9.57 ± 6.42	7.79 ± 5.70	0.00 ± 0.00

Table 5: The table reports contact outcomes for MGA and both controlled ablations. Completed counts denote raw task completion. Loads and violation rates are mean $\pm$ population standard deviation over all ten trials, including failures. None of these three variants records a jam.

Geometric mismatch and evidence scope. Without LRC, only $11/30$ insertions complete and $9/30$ complete safely. Its lower mean tail utilization (0.453 versus 0.814) accompanies a smaller terminal depth (23.2 versus 37.9 mm), so low load alone cannot establish effective contact control. The two unsafe MGA trials occur under Pose-OOD: lateral-force peaks reach 28.31 and 22.81 N against the 20-N limit. PegasusFlow matches its $8/10$ safe completions in this setting and attains lower mean tail utilization (0.630 versus 0.893). Thus, the overall gain does not imply uniform dominance or absolute safety under model mismatch. All settings share one additive-prior configuration and apply execution perturbations to every method. ATACOM inserts in all 30 trials but violates wrench limits in 25. ISSA inserts in 29, with 25 jams and 11 wrench violations (overlapping counts).

Alignment and contact load along the insertion. Figure 12 relates the replayed insertion to lateral-error–depth paths, retaining stalled and unsafe executions. The positional target does not certify contact safety. The full-window criterion also checks orientation, hold time, and load history.

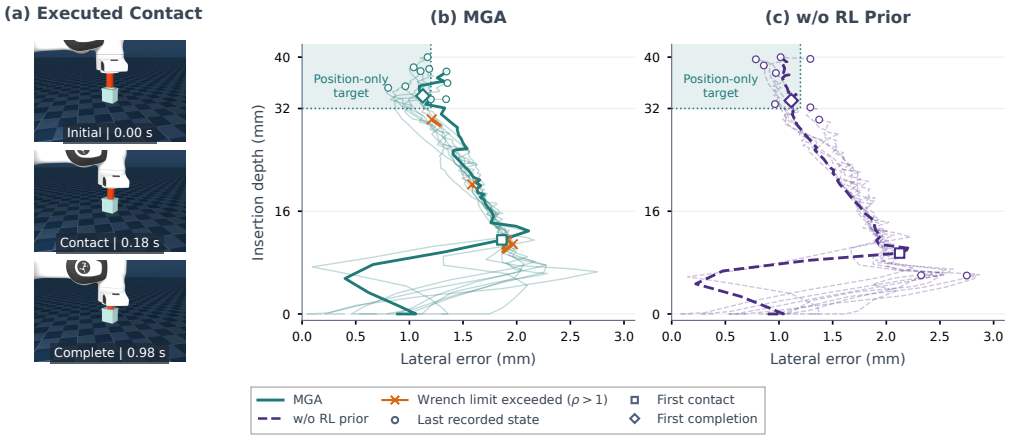


Figure 12: Pose-OOD replay and all ten recorded executions for MGA and w/o RL prior, with seed 0 emphasized. Circles mark endpoints and diamonds first completion. Load exceedances remain visible. The shaded positional target uses lateral error $\leq$ 1.2 mm and depth $\geq$ 32 mm, separately from the orientation, hold-time, and full-window safety conditions.

G.3 Humanoid Push

Force transients and tracking. MGA's mean impact peak is 23.46 ± 4.02 N, compared with 48.63 ± 3.93 N for MPPI, 49.78 ± 8.74 N for DIAL, and 52.73 ± 7.55 N for PegasusFlow. MGA reduces impact but does not minimize every tracking metric (Fig. 13). The 4-ms physics samples resolve peaks potentially missed by control-step observations.

Constructing continued interaction. In Unjamming, MGA safely completes $10/10$ trials, versus $8/10$ without the prior and $3/10$ without LRC (Fig. 14). All seven no-LRC failures stop because no candidate passes revalidation. Correction sustains interaction under the same execution checks.

Load regulation beyond the fixed-force task. All three Fixed-stance Push variants complete $10/10$ trials. Force tails are 0.503, 0.492, and 0.538 for MGA, w/o prior, and w/o LRC. Prior benefits are task-dependent. Unjamming tail summaries exclude rejected prefixes, which count as failures. Conditional tails from eight no-prior and three no-LRC runs are not ranked against MGA's ten-run estimate.

Motion before completion or rejection. Figure 15 connects the stopping events to the box motion. Translation and yaw must be corrected together. A bounded interaction force with unresolved object error does not constitute task completion. The paired paths make this distinction visible while retaining the actual rejection endpoint of w/o LRC. Together with the all-seed continuation counts,

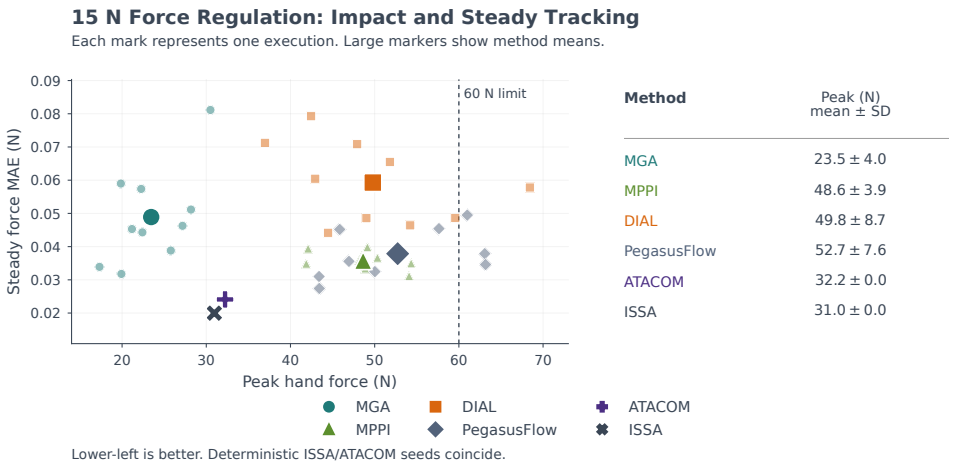


Figure 13: The plot compares impact and steady-state error for 15 N Force Regulation. Small points denote seeds, and large markers denote method means. All methods share physical sampling.



Figure 14: Unjamming traces end at completion or safety-triggered rejection, without absorbing padding. Rejected actions are not executed.

this supports constructing useful accepted sequences rather than interpreting safe rejection alone as successful control.

H Proposal and Refinement Details

This section specifies the proposal protocols in Section 3.1. We fix the current state s^0 and context c. The index h denotes local physical rollout time, and m indexes samples. Solver and dense coordinates follow Appendix F.

Proposal coordinates and reuse. Learned horizons are mapped to the solver coordinates. Gaussian proposals use the DIAL-style node-dependent annealing scale around $\check{U}_k$, with the first node fixed. Control clipping and any enabled trust map are applied before evaluation, so this implemented proposal is not the unmodified isotropic kernel in Eq. (11). In replacement integration, learned horizons occupy M_ψ slots in the common M-candidate bank. They are drawn once per solve and their coordinate and trust maps are reapplied at each reverse step. Gaussian proposals are refreshed at every step.

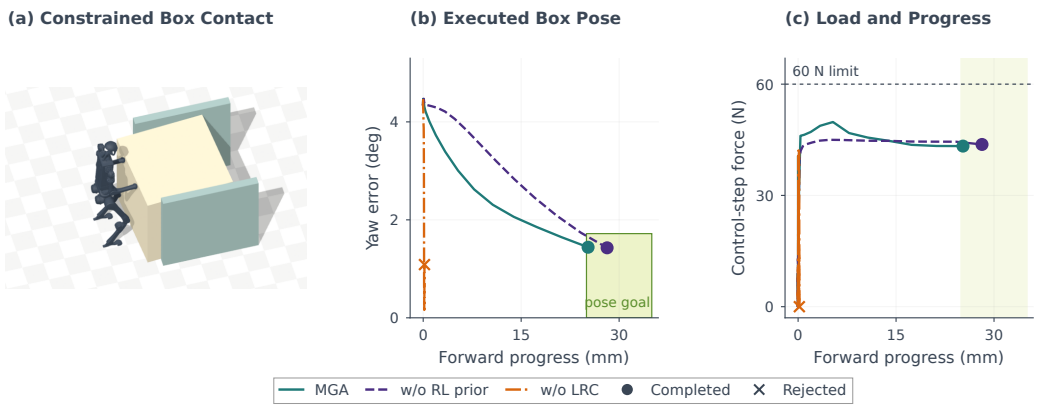


Figure 15: The panels show simulated Unjamming behavior for MGA, w/o RL prior, and w/o LRC. Native imagery provides task context. The plotted motion and force are from the matched executions, ending at completion or rejection without absorbing padding. Force is synchronized with the 20-ms control-step pose records. Physical impact and safety analyses elsewhere retain the finer 4-ms samples. The position–yaw target is a task-completion condition, not a complete safety certificate.

In additive integration, $M_\psi = 0$ in this refinement bank. The Gaussian search starts from the shifted incumbent with the same random stream and realization context as the no-prior variant. After its refinement and execution revalidation, one deterministic and eight stochastic policy horizons are evaluated separately. The best admissible expert competes with the selected Gaussian-path output under the acceptance rules below. These expert horizons do not enter the common softmax or undergo that solve's geometric shaping and LRC. Their evaluations are additional to M.

Optional constraint-aware policies. In replacement integration, the learned budget M_ψ can be split across an RL policy and a stateful constraint-aware expert such as ATACOM. For the latter, let b_m^h be its free action and ζ_m^h its hypothetical slack state:

$$b_m^h \sim \pi_{\mathrm{at}}(\cdot \mid o_m^h), \quad (u_m^h, \zeta_m^{h+1}) = \mathcal{T}_{\mathrm{at}}(s_m^h, b_m^h, \zeta_m^h), \quad U_m^{\mathrm{at}} \sim Q_{\mathrm{at}}(\cdot \mid s^0, \zeta^0, c). \tag{36}$$

Each rollout evolves its own slack copy initialized at ζ^0. The persistent slack is advanced only after real execution. Expert slots replace other stochastic slots rather than increasing M.

Initialization and localization. Let U^ψ denote the deterministic policy rollout. The reverse process starts from $U_K = \sqrt{\bar{\alpha}_K} U^{\mathrm{init}}$, where

$$U^{\mathrm{init}} = \lambda_{\mathrm{shift}} U^{\mathrm{shift}} + (1 - \lambda_{\mathrm{shift}}) U^\psi, \quad \lambda_{\mathrm{shift}} \in [0, 1]. \tag{37}$$

This initialization applies to replacement integration. Additive integration uses $U^{\mathrm{init}} = U^{\mathrm{shift}}$ and does not localize Gaussian proposals around the policy. At the first replan, U^{shift} is zero-initialized. When enabled in replacement integration, proposal localization maps sequences into

$$\left\{ U : \min_{U^{\mathrm{ref}} \in \mathcal{C}} \frac{\|U - U^{\mathrm{ref}}\|_2}{\sqrt{d_U}} \le \epsilon_{\mathrm{trust}} \right\}, \tag{38}$$

where U^{ref} ranges over the reference set $\mathcal{C} = \{U^\psi\}$, optionally enlarged by a deterministic expert rollout, and d_U is the number of optimized scalar coordinates. The trust map is also applied after transport, before residual correction. It does not guarantee that anchors or corrected sequences remain in this set.

Weight normalization. Equation (14) uses $\beta_k = \tau \sigma_{J,k}$, where $\tau > 0$ is the sampling temperature and $\sigma_{J,k}$ is the cost standard deviation over all N_k evaluated candidates, replaced by one below the numerical threshold 10^{-6}. Subtracting the clean iterate's cost from every softmax logit, as in the implementation, leaves the normalized weights unchanged.

Encoded geometry and modulation. For $\Phi(U) := \|C^{\mathrm{real}}(U)\|_2^2/(2n)$, where n is the residual dimension, let $a_j \in \mathbb{R}^{d_u}$ be node j's block of $\nabla_U \Phi(\check{U}_k)$. Select the K_a blocks with largest Euclidean norm, where $K_a = \min\{K_{\mathrm{configured}}, N_u\}$, and use $a_j^\top/(\|a_j\|_2+\epsilon)$ as the rows of $A_k^{\mathrm{node}} \in \mathbb{R}^{K_a \times d_u}$. Here $\epsilon > 0$ is the numerical stabilizer. A zero block therefore remains a zero row. The sequence-space operator is $A_k = I_{N_u} \otimes A_k^{\mathrm{node}}$, with $G_k = I$ and $\lambda_k = 1 + \epsilon$. At $C^{\mathrm{real}} = 0$, this energy gradient vanishes even when $\mathsf{J} \neq 0$, so Lemma 2 alone does not imply contraction of every true chart-normal response.

The task-dependent implementation uses $d_k^{\mathrm{impl}} = d_k + \Gamma(\chi_k \mathsf{M}_k d_k - d_k)$ and transports the clean estimate $\check{U}_k + d_k^{\mathrm{impl}}$. The diagonal matrix Γ repeats the task's action-coordinate reliability gates across the N_u nodes. Each gate lies in $[0, 1]$, and an ungated clean block uses one. The scalar χ_k is the configured geometric gain multiplied by the adaptive feasibility factor, which is one when adaptive modulation is off. The normalized core analyzed in the main text sets $\Gamma = I$ and $\chi_k = 1$. For $\Gamma = \gamma I$, $0 \leq \gamma \leq 1$, and $\chi_k = 1$, the filter's eigenvalues become $1 - \gamma + \gamma\lambda_k/(\lambda_k + \sigma_i^2)$, where σ_i are the singular values of A_k, so nonexpansiveness and kernel preservation remain valid. Arbitrary coordinate-dependent Γ or gain χ_k need not inherit these properties.

Residual correction. Equation (17) acts on the clean sequence after transport and any enabled trust mapping. If repeated, it recomputes the full chart Jacobian and residual at each corrected sequence. Task-dependent reliability can also modulate this correction. Theorem 1 concerns the unmodulated Jacobian correction with $0 < \eta \leq 1$. Its correction-only bound still holds for a trust-mapped input on the same affine branch, but the composed identity from $\check{U}_k$ requires the stated normalized transport. Subsequent task projections and execution acceptance are separate operations and are not covered by these local contraction claims.

Execution revalidation and fallback. Each enabled acceptance check reevaluates the refined sequence and incumbent from the current measured state. The task predicate checks predicted normal-force violations for scanning, force/torque violations and jamming for insertion, and force, contact-validity, and balance violations for humanoid pushing. Their definitions follow Appendix C. If both performance candidates pass, refinement must improve the rollout score beyond the configured tolerance without worsening the task's protected risk components beyond their tolerances. If only refinement passes, it can replace an invalid incumbent without a score improvement. Otherwise a valid incumbent is retained. When neither passes, or a task recovery override is active, the controller considers the task's emergency candidates.

Additive experts first pass the same task predicate and any applicable reliability checks. The best-scoring admissible expert is tested against the already selected Gaussian-path output. Its adoption requires acceptance and strict score improvement on ordinary replans. Task-owned recovery transitions use their dedicated rules and cannot be overwritten by a policy score win. Rejection retains the Gaussian-path decision. For the humanoid execution context, failure of all revalidated continuations stops the run before executing the rejected action. These checks explain the reported execution rejections but do not certify physical safety under model mismatch.

I HYPERPARAMETERS

The settings below follow saved configurations and frozen-component metadata. Environment geometry, physical limits, and success definitions appear in Appendix C.

I.1 LEARNED PROPOSAL SETTINGS

All proposal policies remain frozen during evaluation. Table 6 separates training budgets from selected checkpoints. Scanning also uses a separately trained ATACOM proposal expert. The comparisons test deployment of learned behavior, not further training.

The scanning policy trains on rigid cylinder and convex surfaces and compliant cylinder and bumpy surfaces. The insertion policy uses nominal, tight-clearance, and randomized conditions. H1 shares one policy across its three tasks. Checkpoints follow the saved development criteria: realized progress for scanning, the final fixed-budget checkpoint after an independent health check for in-

Setting	Scanning	Peg Insertion	Humanoid Push
Policy optimizer	PPO	PPO	PPO
Hidden layers	4×32	2×16	2×16
Recorded training budget	2 M	200 k	200 k
Selected PPO checkpoint	1 M	200 k	Final
ATACOM proposal expert	Yes	No	No
Selected ATACOM checkpoint	500 k	–	–
Policy updates in evaluation	None	None	None

Table 6: The table lists frozen proposal settings, with budgets measured in training environment steps. The ATACOM rows describe a proposal expert. The separate baseline is described in Appendix D.

sertion, and the final fixed-budget policy for H1. The separate insertion ATACOM baseline also uses its final 200-k-step tangent-policy checkpoint.

I.2 Model-Based Annealing Settings

Table 7 separates evaluation length from planning horizon. Table 8 records the proposal sources needed to interpret rollout work.

Setting	Scanning	Peg Insertion	Humanoid Push
Evaluation intervals	100	64	100
Control interval (s)	0.02	0.02	0.02
Dense-grid intervals (H_{sample})	16	8	16
Node intervals (H_{node})	4	4	4
Refinement-bank budget M	64	64	64
Initial refinement iterations	10	4	10
Subsequent refinement iterations	2	2	2
Cost-scaled temperature τ	0.06	0.06	0.06

Table 7: The table lists the recorded MGA inference settings. Peg insertion retains the full 64-step window after completion. Other tasks use their documented termination rules. Sampling temperature enters $\beta_k = \tau\sigma_{J,k}$ and is not a physical contact parameter.

Recorded setting	Gaussian	PPO	ATACOM	Integration
Scanning	48	8	8	Replacement bank
Peg Insertion: all settings	64	8	0	Additive expert
Humanoid Push	64	8	0	Additive expert

Table 8: The table lists the recorded stochastic proposal counts. Deterministic anchors and execution-revalidation queries are additional.

Replacement uses learned slots in the 64-sample bank (Section 3.1), reusing those proposals across reverse steps while refreshing Gaussian samples. Additive integration retains 64 Gaussian samples and separately evaluates one deterministic plus eight stochastic policy horizons after Gaussian refinement and revalidation. These nine horizons are outside the softmax bank and its shaping/LRC updates. Peg ID and OOD use the same additive configuration. The prior comparison therefore retains additional policy rollouts and does not establish equal-total-rollout efficiency. Action-filtering baselines do not inherit a trajectory sampler.

I.3 Realization-Aware Refinement Settings

Task-specific charts (Appendix F) share the operators in Table 9. Feasibility penalties evaluate candidates, and LRC corrects the transported clean sequence.

The **w/o RL prior** variant retains the task's refinement and execution mechanisms. The **w/o LRC** variant retains proposals, geometric shaping, and acceptance checks and does not remove physical

Setting	Scanning	Peg Insertion	Humanoid Push
Linear feasibility coefficient	2	2	2
Quadratic feasibility coefficient	200	80	200
Normalized geometric metric	I	I	I
LRC iterations per correction	1	1	1
LRC gain η	1	1	1
LRC regularization ε	10^{-6}	10^{-6}	10^{-6}

Table 9: The table lists the evaluation and normalized correction settings. Task modulation, trust maps, and execution checks are detailed in Appendix H. The operator alone does not certify safety.

withdrawal. PegInsert uses core-only refinement with learned reliability disabled in all three variants. In scanning and H1, learned reliability defers to task-owned checks, so these ablations do not establish an independent learned-confidence gain or a continuous parameter sweep.

J VISUALIZATIONS

Each row shows five states from one MGA execution, with a fixed camera and native robot assets. We retain all 13 scanning suites, three insertion settings, and three humanoid tasks—including incomplete cases. Quantitative analyses use all evaluation seeds. Each frame reports simulation time t and executed contact force F: surface-normal force for scanning, contact-force magnitude for insertion, and summed hand–box normal forces for humanoid tasks. A dash indicates an unrecorded initial force, not zero force.

J.1 SURFACE SCANNING

Stiffness insets show nominal task-chart fields or uniform contact parameters, not measured deformation. Rigid contact is categorical. White dots locate the displayed command states, not end-effector world positions, and the dashed line marks $\eta = 0.5$.

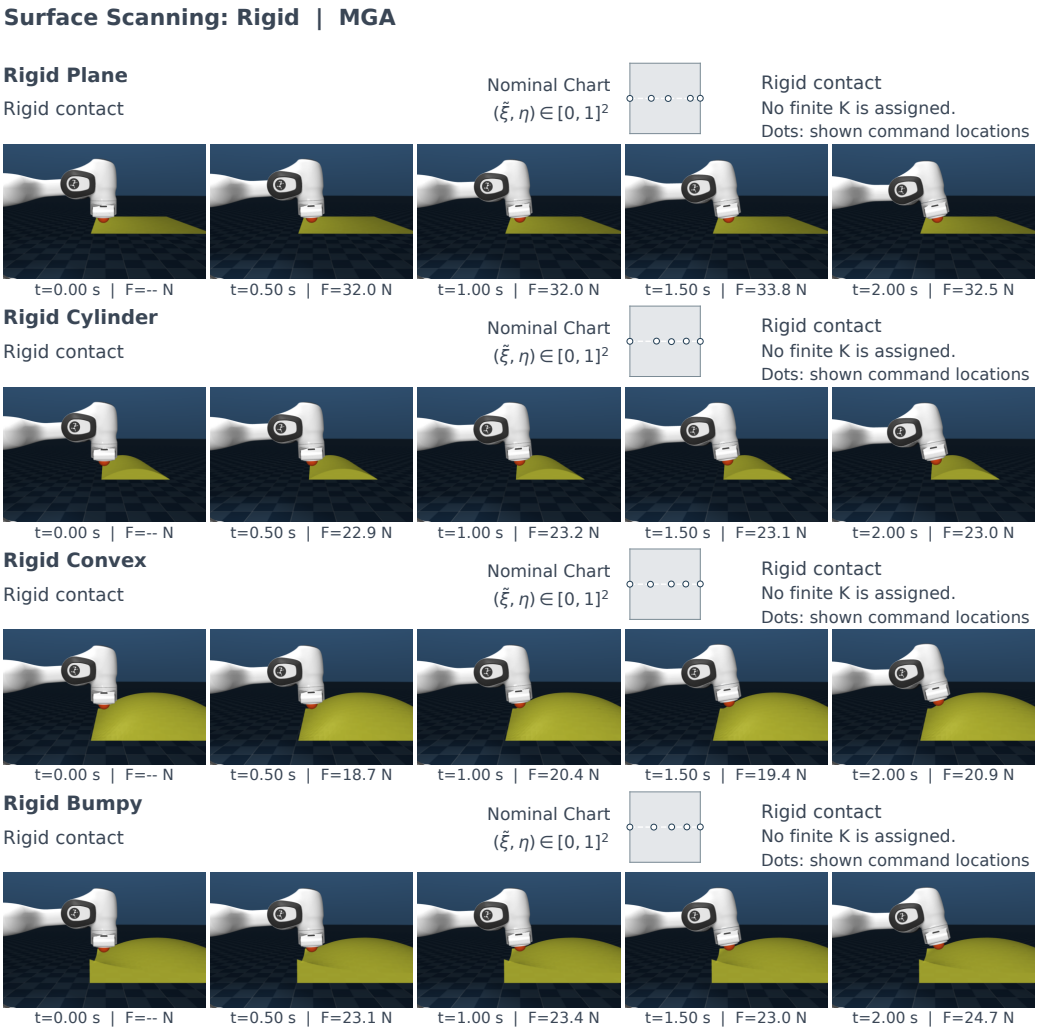


Figure 16: MGA scans the four rigid surfaces: plane, cylinder, convex, and bumpy. Five recorded states per row show the executed robot and tool motion from the same seed, with a fixed camera and common rendering style. The contact inset identifies the rigid regime without a finite-stiffness color assignment.

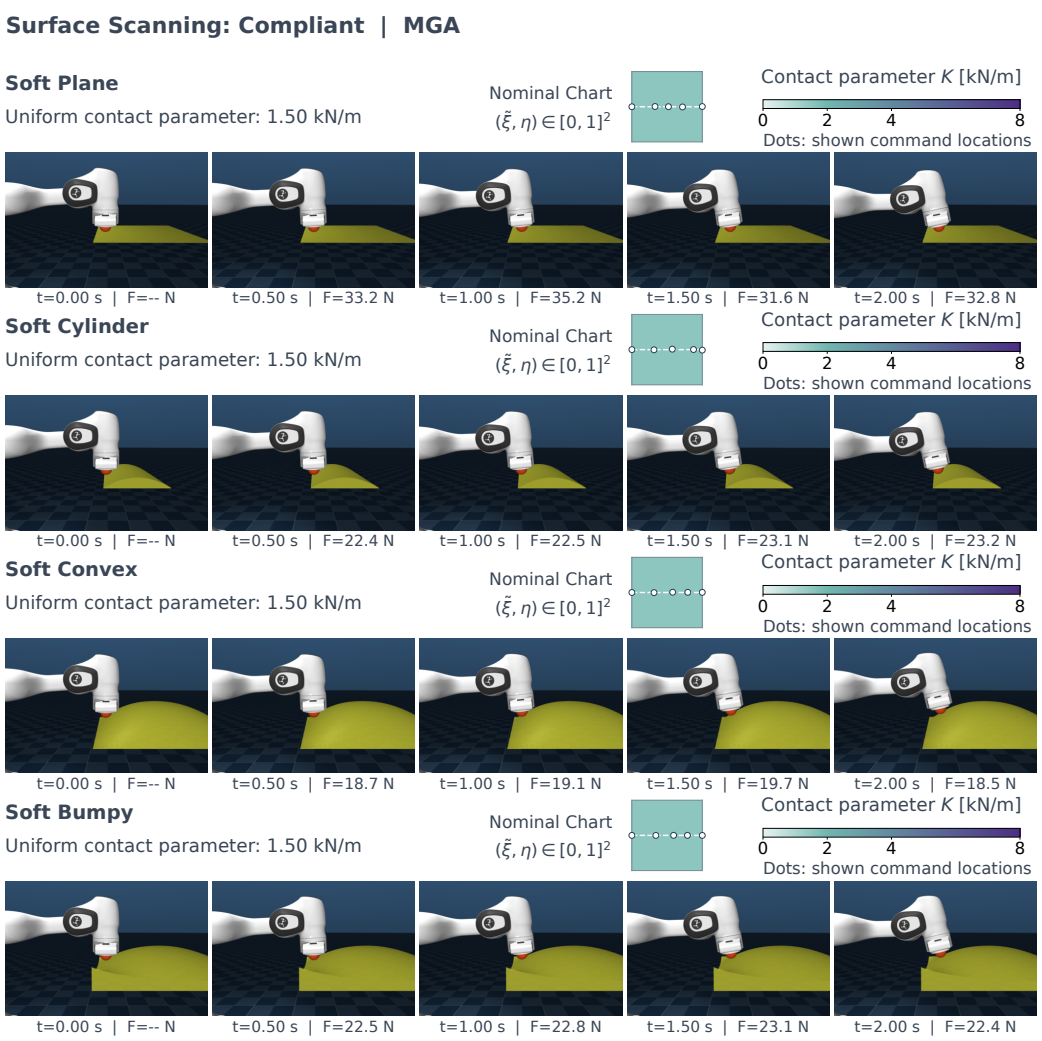


Figure 17: MGA scans the four compliant surfaces. The frames show recorded robot motion against the nominal rendered surface. Mesh appearance alone does not measure deformation, which is quantified separately in Fig. 8. Contact-parameter insets distinguish the compliant setting from the corresponding rigid geometry.

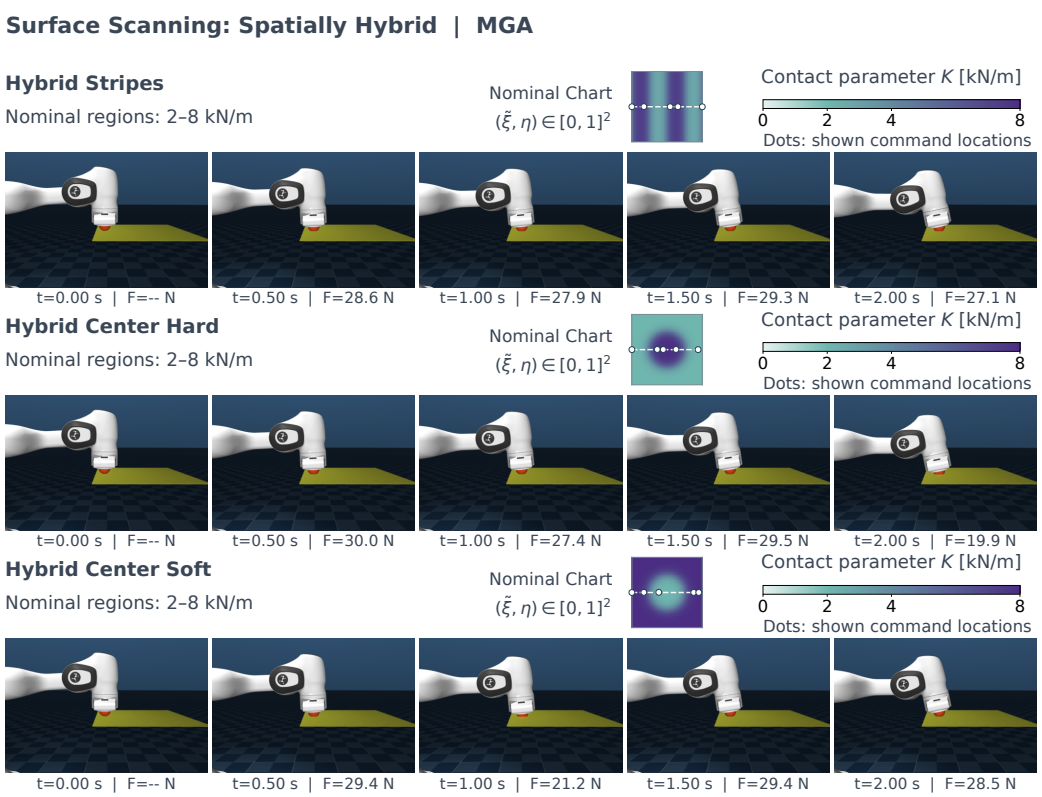


Figure 18: The frames show MGA executions across three planar hybrid stiffness maps: alternating stripes, hard center, and soft center. Material variation changes the contact response without introducing a curved surface. Insets show the nominal task-chart stiffness field. It is evaluated at the commanded surface coordinate, not inferred from the displayed end-effector position. All frames are states from the recorded executions.

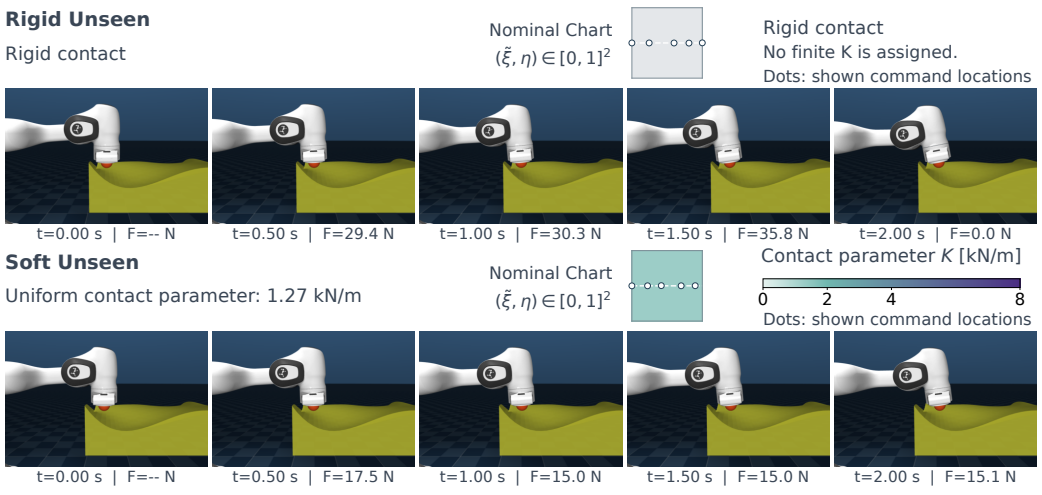


Figure 19: MGA scans the held-out rigid and compliant surfaces. The rigid example is retained despite failing the scanning success criterion, so visual coverage of the benchmark does not select only successful cases.

J.2 Peg Insertion

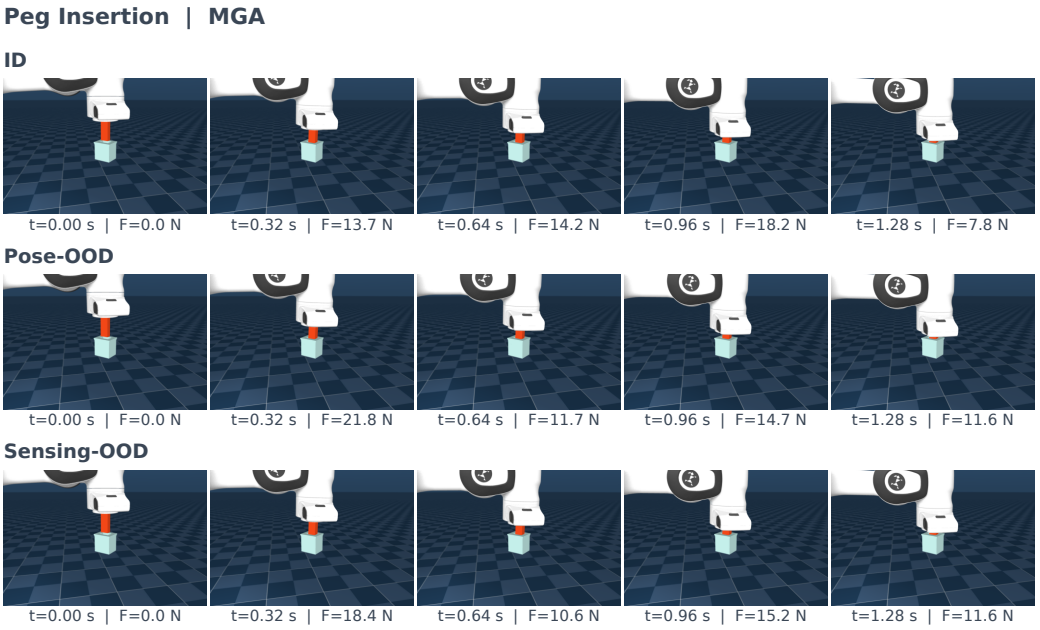


Figure 20: Five chronological pose replays show MGA under ID, Pose-OOD, and Sensing-OOD. Native robot views replay recorded joint configurations without re-simulation. Quantitative plots use the saved task-coordinate signals. The fixed examples all complete safely. The statistical comparisons retain every trial, including the two unsafe Pose-OOD executions.

J.3 Humanoid Push

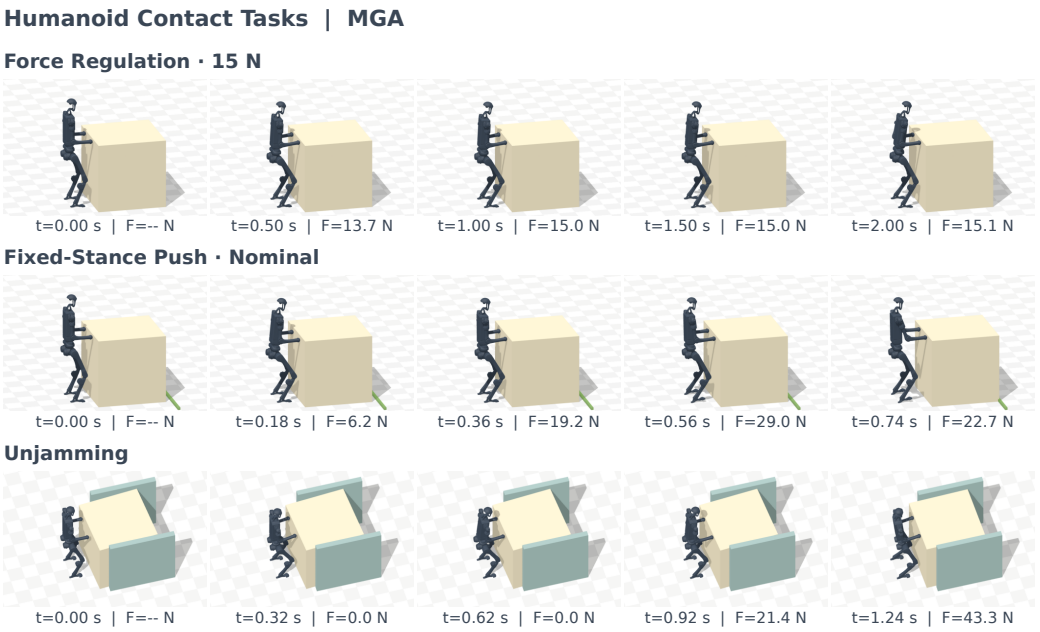


Figure 21: The frames show MGA in Force Regulation, Fixed-stance Push, and Unjamming. Five frames span each execution before terminal padding. Goal markings appear only in the corresponding displacement tasks. These are not walking pushes.