LATENT JACOBIAN STEER: ELICITING CONTACT-RICH OUT-OF-DISTRIBUTION BEHAVIORS FROM PRETRAINED HUMANOID CONTROLLERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Given a pretrained general latent-conditioned whole-body humanoid controller, how can we elicit complex and contact-rich out-of-distribution behaviors? Joint-action exploration enables direct task-specific adjustments but can disrupt coordination learned by the pretrained controller, whereas latent-command exploration retains it but is difficult to steer toward desired physical effects. To combine the complementary strengths of these exploration spaces, we propose Latent Jacobian Steer (LJS), which couples sampling in direct joint-action space with the semantics of the learned frozen controller's input latent space. For a task specified through desired physical effects, we sample task-relevant joint-action perturbations and map them through a damped local inverse of the policy Jacobian to obtain candidate latent commands. These commands are executed through the frozen pretrained policy, while exploration is directed toward the physical outcomes required by the new task. Candidate outcomes are evaluated in simulation and composed into latent-command sequences that accomplish the task. Ablations show that neither action-space nor latent-space exploration alone provides the same capability as their coupling. Across diverse out-of-distribution tasks, test-time searching using this method produces contact-rich behaviors, including hurdling, manipulating articulated objects, pivoting large objects against the environment, and opening doors through non-prehensile contact.

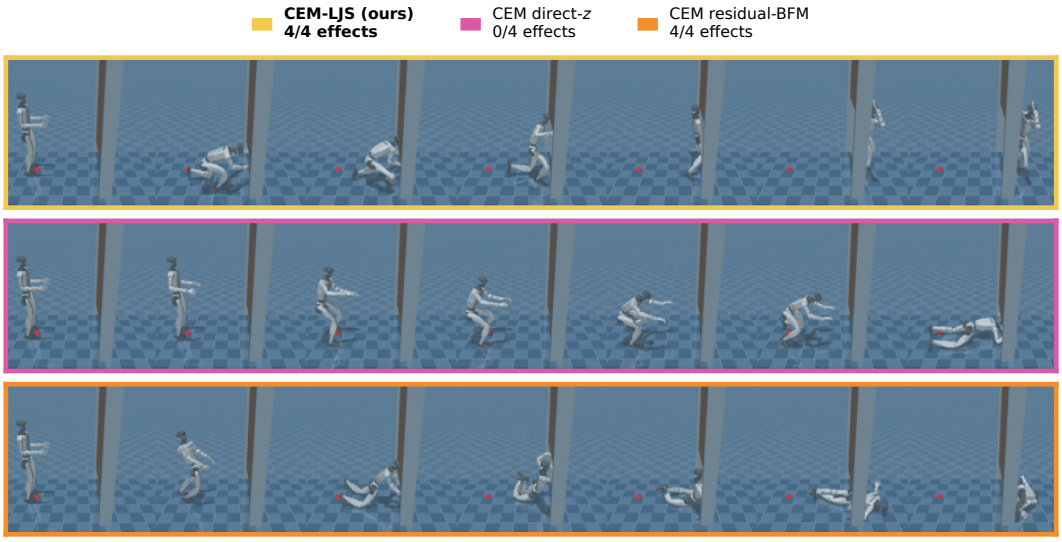


Figure 1: **Shutter traversal: completion and motion quality.** Seven chronological frames (left to right) from a successful **CEM-LJS (ours)** run and the best seeds of baseline runs. CEM-LJS lifts the shutter and crosses on its feet. Direct-z fails palm placement and ends prone. CEM-Residual-BFM completes it but hacks the task and crosses with its torso on the floor.

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.

1 INTRODUCTION

For humanoid robots to become useful, they must be able to physically interact with objects and environments in diverse and often contact-rich ways. Recent advances in motion capture and general motion-tracking have enabled humanoids to reproduce an increasingly broad repertoire of dynamic whole-body behaviors (Chen et al., 2025; Liao et al., 2026). Yet obtaining behaviors for a new contact-rich loco-manipulation task remains considerably more task-specific. Such behaviors often require dedicated human demonstrations or trajectories reconstructed from video through Real2Sim pipelines that recover the relevant object and environment interactions (Dan et al., 2025; Ranawaka et al., 2026).

High-fidelity simulation provides a natural way to generate such behaviors. Existing approaches using simulation either learn new behaviors from task-specific rewards (Sferrazza et al., 2024; Mu et al., 2024) or explicitly generate demonstrations and trajectories for them (Jing et al., 2025; Taouil et al., 2026) More recently, Behavioral Foundation Models such as Meta Motivo and BFM-Zero have instead pretrained general latent-conditioned whole-body controllers that can be prompted towards different motions, goals, and rewards (Tirinzoni et al., 2025; Li et al., 2026b). These approaches inspire our question: can we steer such a frozen controller to produce complex contact-rich behaviors that lie outside its training distribution, without collecting a task-specific motion demonstration or training a new policy?

We hypothesize that much of the whole-body coordination required for these behaviors is already present in a sufficiently general pretrained controller, even if it was never trained on such object-environment interaction data. This is fairly intuitive in hindsight. A human who has learned to move in free space can use these motion priors to play around with objects and the environment. The challenge is how to *steer* it towards a new physical interaction. Direct exploration in joint-action space makes task-specific changes easy to express, for example, moving a hand towards a desired contact, but it can disrupt the coordination learned by the controller. Exploration in its latent space preserves this coordination, but it is difficult to know which latent perturbation will produce a desired physical effect. Our key observation is that these two spaces are locally connected by the frozen policy itself: its Jacobian describes how perturbations in latent commands change the resulting joint actions.

Based on this observation, we introduce **Latent Jacobian Steer (LJS)** (Figure 2), which couples precise exploration in joint-action space with coherent execution through the latent-conditioned pretrained controller. Given a task-relevant joint-action perturbation, LJS uses a damped local inverse of the policy Jacobian to obtain candidate latent commands, which are then executed and evaluated in simulation. Complex tasks can then be described through simple sequences of physical effects: opening a door, for example, can be specified as (1) reaching and grasping the handle, (2) manipulating it to release the latch, and (3) pushing the door about its hinge. In this way, LJS provides a means of eliciting and collecting new contact-rich behaviors from a frozen whole-body controller without requiring a task-specific motion demonstration, a complete reference trajectory, or training a new task policy.

2 RELATED WORKS

From motion tracking to contact-rich behavior discovery. General trackers such as TWIST (Ze et al., 2025) and SONIC (Luo et al., 2026) provide broad whole-body motion repertoires, while PGMT (Li et al., 2026a) and GigaBrain-WBC-0.5 (Cheng et al., 2026) incorporate terrain perception and environment interaction. Motion generators such as OMOMO (Li et al., 2023) and CHOIS (Li et al., 2024) produce human–object interactions from object trajectories or language and spatial guidance, using captured interactions as training data. For a new contact-rich task, however, the movement that realizes the required interaction must still be obtained. Real2Sim pipelines reconstruct it from demonstrations (Dan et al., 2025; Ranawaka et al., 2026); contact-implicit trajectory optimization (Posa et al., 2014) and logic-geometric programming (Toussaint, 2015) optimize motions and manipulation decisions. More recently, MotionDisco (Taouil et al., 2026) combines motion search with trajectory optimization, while OmniContact (Yu et al., 2026) chains skills through contact-flow representations. We investigate whether these interactions can instead be discovered by

steering the existing coordination of a frozen whole-body controller, without supplying a reference motion or training a new task policy.

Reusing and steering pretrained behaviors. Latent skill methods such as SPiRL (Pertsch et al., 2020), OPAL (Ajay et al., 2021), and SkiMo (Shi et al., 2023) provide reusable primitives for downstream learning or planning. METRA (Park et al., 2024b) and HILP (Park et al., 2024a) learn behavioral spaces before downstream tasks are specified, while forward-backward representations (Touati & Ollivier, 2021) support reward-based policy inference without retraining. This direction underlies whole-body behavioral models including Meta Motivo (Tirinzoni et al., 2025), BFM-Zero (Li et al., 2026b), and FB-MEBE (Hu et al., 2026). Their behavioral coverage provides a useful starting point, but finding commands for precise new interactions remains difficult. Adaptation methods address this through parameter updates, skill-based learning, or action corrections, as in OpenVLA (Kim et al., 2025), SUPE (Wilcoxson et al., 2025), and residual RL (Johannink et al., 2019). Closer to our setting, FlowDAgger (Murray et al., 2026), UniSteer (Lu et al., 2026), and Flow Reversal Steering (Tang et al., 2026) map human or VLM corrections into the noise space of frozen generative policies. These works connect action-space guidance to latent execution. LJS instead discovers useful corrections from simulator-evaluated physical outcomes, without expert corrective actions or a learned steering policy.

Searching through the policy's local inverse. CEM (Rubinstein, 1999), iCEM (Pinneri et al., 2020), MPPI (Williams et al., 2017), and DIAL-MPC (Xue et al., 2025) search through simulated outcomes; SkiMo (Shi et al., 2023) and BFM-Zero (Li et al., 2026b) also demonstrate planning over learned behavioral spaces. Our question concerns how to construct useful proposals within a frozen controller. Structured exploration methods such as gSDE (Raffin et al., 2022), LATTICE (Chiappa et al., 2023), and DEP-RL (Schumacher et al., 2023) organize perturbations for coordinated control, while Safe Mutations (Lehman et al., 2018) and Policy Manifold Search (Rakicevic et al., 2021) use output sensitivities to guide exploration. Drawing on local inverse mappings used in damped inverse kinematics (Wampler, 1986) and operational-space control (Khatib, 1987), LJS couples joint-action proposals to latent execution through the policy Jacobian. CEM searches joint-command changes, a damped inverse maps them into latent perturbations, and simulator rollouts evaluate their physical effects. The Jacobian describes the frozen actor's local response, not contact dynamics. This coupling lets joint-action coordinates structure the search while the pretrained controller generates every executed command.

3 LATENT JACOBIAN STEER

Latent Jacobian Steer (LJS) couples direct joint-action and latent search spaces. We search for task-relevant changes in joint-action space, map them to changes in the latent command through the local Jacobian of the frozen policy, and evaluate the resulting behavior through physics simulation. Figure 2 gives an overview.

Problem Formulation. We consider humanoid control in a simulator as a finite-horizon Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{U}, P, \rho_0, T)$, where $\mathcal{S}$ is the state space, $\mathcal{U}$ is the joint-command space, P is the transition dynamics including the low-level controller, ρ_0 is the initial-state distribution, and T is the task horizon. We specify the task objective separately via the physical-effect sequence $\mathcal{T}$, defined below, whose scores guide the search and whose acceptance conditions determine task completion. The state s_t includes the robot and environment configurations and velocities, together with the controller memory needed to make the transitions Markovian. For the 29-DOF humanoid considered here, $u_t \in \mathcal{U} \subseteq \mathbb{R}^{29}$ specifies joint-position targets, realized through a joint-space PD controller. We have a frozen pretrained policy $u_t = \pi_\theta(h_t, z_t)$, where h_t contains the robot's proprioceptive observations and action history, and $z_t \in \mathcal{Z}$ is its latent command. Now, for our particular purpose, to demonstrate the adaptation of the pretrained policy over interesting OOD tasks, we begin by concretizing *how to define a task itself*. It is worth noting that we chose this as our task definition for ease of expressing various tasks, and we are not proposing it as a canonical way to define loco-manipulation tasks. We define a task as an ordered sequence of physical effects $\mathcal{T} = (E_1, \ldots, E_K)$. Each effect E_k specifies a physical measurement ϕ_k, a desired target or target region Y_k, and a non-negative loss ℓ_k. For a trajectory segment ξ, its physical-effect score is $c_k(\xi) = \ell_k\big(\phi_k(\xi), Y_k\big)$. The measurement may describe body or object displacement, orientation, contact forces, or conditions that must hold over the segment. When an effect requires several conditions simultaneously,

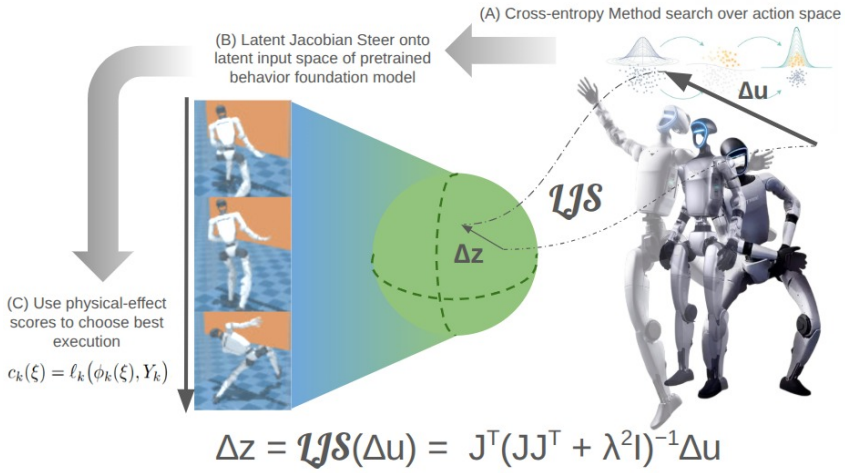


Figure 2: **Latent Jacobian Steer overview.** (A) CEM proposes joint-action perturbations Δu. (B) A damped inverse of the frozen policy Jacobian maps each proposal to a latent change Δz; the updated latent command is normalized and simulated through the pretrained controller. (C) Physical-effect scores rank the rollouts and guide CEM updates. The best admissible candidate is executed for a short window before replanning. The actor remains frozen throughout.

we combine their losses: $c_k(\xi) = \sum_{j=1}^{m_k} w_{k,j}\ell_{k,j}\big(\phi_{k,j}(\xi), Y_{k,j}\big), w_{k,j} > 0$. We also specify an acceptance condition $A_k(\xi)$ for each effect. By default, an effect is accepted when all its component losses are zero; separate acceptance tolerances can be specified where needed. This distinguishes the score used to guide exploration from the condition used to decide whether the physical effect has been achieved. The task is satisfied when the executed trajectory admits ordered boundaries $0 = t_0 \le t_1 \le \cdots \le t_K \le T$ such that $A_k\big(\xi_{t_{k-1}:t_k}\big) = 1; k = 1, \ldots, K$. Any condition that must persist across multiple effects is included in each relevant effect. Thus, we specify the physical outcomes required at successive stages, while leaving the controller and search to find the movements that realize them. The search has access to privileged simulator state and contact information to evaluate the desired physical effects; the pretrained policy continues to receive its original observations. Our goal is to find a sequence of latent commands that satisfies these effects without changing θ. Here, OOD tasks require contact interactions and physical effects that are not represented during the controller's pretraining.

The choice of pretrained controller is important for LJS. A latent-conditioned policy or general motion tracker is not necessarily suitable simply because it supports diverse behaviors: its latent space must also be meaningfully steerable toward downstream objectives. For our main experiments, we use BFM-Zero (Li et al., 2026b), which learns a shared latent space through unsupervised reinforcement learning with forward-backward representations and can be prompted using motions, goals, or rewards without task-specific retraining. This is particularly useful in our setting, where we specify the desired physical effect without providing a reference motion that realizes it. We therefore use LJS to explore how this latent space can be steered toward new contact-rich behaviors.

3.1 Our choice of latent-conditioned pretrained controller, BFM-Zero

BFM-Zero (Li et al., 2026b) learns a family of whole-body behaviors through a latent-conditioned policy $\pi_\theta(h, z)$. Its forward-backward (FB) representation relates latent commands to the future states the policy visits, allowing motions, goals, and rewards to prompt the same controller without task-specific retraining. This makes it useful for our setting, where we specify a desired physical effect without supplying a reference motion that realizes it.

For the released G1 checkpoint used here, the latent command lies on the sphere $\mathcal{Z} = \{z \in \mathbb{R}^{256} : \|z\|_2 = 16\}$. The actor produces 29 joint-position targets at 50 Hz. LJS searches these latent commands while keeping the actor frozen; the forward and backward mappings are not required during search. We defer the FB formulation and its prompting mechanisms to Appendix A.1.

We propose Latent Jacobian Steering (LJS) with the following components as a solution to the problem of steering such a controller to discover complex OOD behaviors:

3.2 Defining the physical effects for a contact-rich OOD task

We define a task through a small sequence of measurable physical outcomes. For shutter traversal, these are placing the palms beneath the shutter, lifting it, moving the head and torso through the opening, and clearing it with the entire body. Each effect specifies what must happen while leaving the movements that realize it to the search. For example, lifting can require a minimum shutter height and sufficient upward support, with a score penalizing shortfalls and a separate acceptance condition determining completion.

We construct these effects stage-wise, using an LLM as a coding assistant to translate intuitive task descriptions into simulator measurements and scores. We inspect rollouts and refine definitions that admit unintended outcomes. Definitions remain human-specified and fixed during each search; revisions are rerun from the preceding effect's terminal state. We defer the mathematical score definitions, task-specific sequences, and acceptance conditions to Appendix C.

3.3 Coupling of the latent and direct action space

Once the physical effect for a particular stage of a task is defined, our goal is to find latent commands that satisfy it. We do this by searching in joint-action space using CEM (Rubinstein, 1999) and pulling the proposed changes back into the latent space through the frozen policy's local Jacobian.

How is a joint-action change pulled back into the latent space? Let $u(h, z)$ denote the joint-position targets produced by the frozen policy, including its output scaling and latent normalization. At the current observation history h and latent command z, we compute

$$J(h, z) = \frac{\partial u(h, z)}{\partial z} \in \mathbb{R}^{29\times 256}.$$

This is the policy Jacobian: it describes how the current joint targets change with the latent input while h is held fixed. For a small latent change,

$$u(h, z + \Delta z) - u(h, z) \approx J(h, z)\Delta z.$$

Thus, given a proposed joint-action change Δu, we seek a latent change that approximately realizes it. Since J is rectangular and may have weakly responsive directions, we use a damped least-squares inverse:

$$\Delta z^\star = \arg\min_{\Delta z} \|J\Delta z - \Delta u\|_2^2 + \lambda^2\|\Delta z\|_2^2, \qquad \lambda > 0.$$

Setting the gradient to zero gives

$$(J^\top J + \lambda^2 I)\Delta z^\star = J^\top \Delta u,$$

and hence

$$\Delta z^\star = \underbrace{J^\top (JJ^\top + \lambda^2 I)^{-1}}_{A} \Delta u.$$

The damping limits latent changes along directions in which the policy is insensitive. We then form the candidate

$$z' = \Pi(z + A\Delta u), \qquad \Pi(v) = 16\frac{v}{\|v\|_2}.$$

The Jacobian includes differentiation through Π, so the local change accounts for the controller's spherical latent domain. This mapping approximates the policy's immediate output response; we evaluate the resulting physical movement through simulation.

How do we search for joint-action changes? We adapt the CEM implementation in Hydrax (Kurtz, 2024) to search over joint-action perturbations. At each planning step, we initialize a zero-mean diagonal Gaussian and divide the candidate budget across several CEM rounds. Each sampled perturbation is mapped into a latent command: $\Delta u_i^{(q)} \sim \mathcal{N}\big(\mu_q, \mathrm{diag}(s_q^2)\big)$, $z_i^{(q)} = \Pi\Big(z + A\Delta u_i^{(q)}\Big)$. The latent anchor z and pull-back map A remain fixed across rounds. We simulate the candidates, rank

them by their physical-effect scores, and refit the Gaussian to the best admissible joint-action proposals. Each round includes its current mean, so the first round also tests the unchanged command. A minimum standard deviation retains exploration; if no candidate is admissible, the distribution is left unchanged. We defer the full CEM updates and elite-selection rules to Appendix A.2.

How do we score and execute a candidate? We simulate each candidate for H control steps from the same saved simulator state and controller history, holding its latent command fixed while the frozen actor receives updated observations. We rank candidates by their mean physical-effect score over the latter half of the rollout, allowing time for the response to develop. Candidates violating full-segment constraints are rejected and excluded from the elite set.

After all CEM rounds, we execute the lowest-scoring admissible candidate for $L \leq H$ steps. Its latent command becomes the new anchor; we recompute the Jacobian and restart CEM with zero-mean perturbations. We advance only when the executed segment satisfies the current effect's acceptance conditions. Search ends when all effects are accepted, the budget is exhausted, or no admissible candidate is found. The exact rollout objective and search parameters are provided in the appendix.

4 EXPERIMENTS

We evaluate whether LJS can elicit contact-rich behaviors from a frozen whole-body controller, whether coupling joint-action proposals to latent execution improves task completion, and how the choice of search optimizer affects the result. All experiments use the frozen BFM-Zero G1 controller in MuJoCo, without task-specific policy training or reference motions.

4.1 CONTACT-RICH TASKS

We consider three tasks that require different combinations of contact establishment, whole-body support, object manipulation, and recovery. Each task is defined through an ordered sequence of physical effects.

Hurdle traversal. The humanoid must cross a rigid hurdle using its body for support. It establishes palm and leg contacts, moves into a supported straddling configuration, transfers its body across the obstacle, places both feet on the far-side floor, and returns to standing. The ten effects require alternating support configurations, leg clearance, and recovery after crossing.

Shutter opening and traversal. The humanoid must place both palms beneath a vertically sliding shutter, lift it, move its head and torso through the opening, and clear the panel with its entire body. These four effects combine precise palm placement and orientation with upward support and crouched whole-body movement.

Door opening while carrying a cube. The humanoid approaches a cube on a stand, grasps its opposing side faces, lifts it clear, and carries it to a closed door. It then uses the grasped cube to depress the handle and push the door open. The six effects require retaining the cube while manipulating another object; the final effect requires a door angle of at least $0.95\,\mathrm{rad}$, bilateral palm contact with the cube, and contact between the cube and door.

4.2 DEFINING THE PHYSICAL-EFFECT SCORES

We construct effects stage-wise, beginning with an intuitive description of the required outcome. An LLM coding agent helps translate this description into simulator measurements, such as body-object distances, object orientation, contact forces, clearance, and velocity. These measurements define a score for ranking candidates and a separate acceptance condition for advancing to the next effect. They specify physical outcomes without prescribing the complete movement that realizes them.

We inspect the resulting rollouts and revise definitions that admit unintended solutions, rerunning the current effect from the preceding accepted effect's terminal state. Every definition is human-reviewed and held fixed during evaluation; the LLM does not observe the search online or select commands. This back-and-forth required less than five hours to finalize a task's effects. All methods are evaluated with the same final task definitions. The mathematical definitions and an example Human–AI interaction card (Feng et al., 2026) are provided in Appendix C; practical guidance is given in Appendix D.

Table 1: **Contact-rich task completion.** Mean accepted physical effects over three seeds, relative to the task total.

Method	Hurdle	Shutter	Door-cube
CEM-LJS	**7.00/10**	**4/4**	**5.33/6**
CEM direct-z	5.67/10	0/4	3.33/6
CEM residual-BFM	4.67/10	2.67/4	2.00/6
Mixed-Gaussian LJS	3.33/10	0/4	2.33/6

4.3 Baselines and Evaluation

Our main comparisons keep CEM as the optimizer and change where proposals are made and how they are executed. We additionally ablate CEM while retaining the LJS coupling.

CEM-LJS. Our method searches over joint-action perturbations and maps each proposal into a latent command through the damped policy-Jacobian inverse. CEM fits its proposal distribution to the best simulated candidates, and the selected command is executed through the frozen actor.

CEM direct-z. We apply CEM directly to the 256-dimensional latent command, following the latent-optimization approach demonstrated by BFM-Zero (Li et al., 2026b). This retains the optimizer and frozen-policy execution while removing the joint-action proposal space and Jacobian pull-back.

CEM residual-BFM. CEM searches for additive corrections to the frozen actor's joint-position targets. This retains joint-action proposals but removes the requirement that they be realized through latent commands, giving the search direct authority to override the actor's output.

Mixed-Gaussian LJS. This optimizer ablation retains the Jacobian pull-back and replaces CEM with Gaussian proposals. Below a physical-effect score threshold, finite-difference simulator probes estimate a targeted correction. The candidate population combines proposals around that correction and around zero, together with directional trials and the unchanged command.

We report the mean number of accepted effects relative to each task's total across three seeds. Individual runs are reported in Appendix E. Search cost counts simulated candidate control steps, including rejected candidates. For each effect k, we estimate the expected runtime (ERT) in these steps (Hansen et al., 2016):

$$\widehat{\mathrm{ERT}}_k = \frac{\sum_{i=1}^{3} C_{i,k}}{\sum_{i=1}^{3} \mathbf{1}\{\text{run } i \text{ reaches effect } k\}}.$$

Here, $C_{i,k}$ counts steps from task initialization to the first acceptance of effect k, or to termination if it is not reached. This estimates the cost of reaching the effect under independent restarts from the task's initial state, charging unsuccessful attempts as well. No finite estimate is available when all runs fail to reach an effect. For shutter, the prediction horizon is 20 control steps, each selected command is executed for five steps, and each effect allows at most 30 decisions. For door-cube, all methods evaluate 1,024 candidates per decision and execute each selected command for ten control steps. The first four effects use a 40-step prediction horizon and a 15-decision cap; the final two use a 20-step horizon and a 30-decision cap. Both door-cube settings allow 614,400 simulated candidate control steps per effect. Hurdle uses 40-step prediction horizons with a 30-decision cap for the first five effects, and 100-step horizons with a 15-decision cap for the remaining five. Commands execute for 20 steps, except effects 3–5, which use five steps. Search stops when the task is complete, the current effect exhausts its budget, or no admissible candidate is available.

4.4 Results

Contact-rich task completion. CEM-LJS achieves the highest mean effect completion on all three tasks (Table 1). It completes hurdle traversal in two of three seeds; no baseline completes the task. Direct-z and residual-BFM reach at most seven hurdle effects, while Mixed-Gaussian LJS reaches at most five. On shutter, CEM-LJS completes all three seeds, compared with one for residual-BFM and none for the other methods. On door-cube, it completes two seeds, compared with one for direct-z

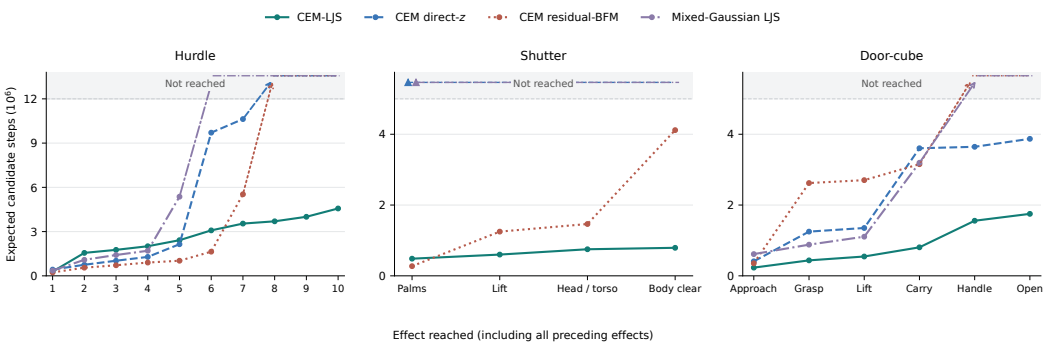


Figure 3: **Search cost to reach each physical effect.** Estimated expected candidate control steps across three seeds per task, including unsuccessful attempts. Each target includes all preceding effects; lower is better. Panels use different vertical scales. Hurdle effects follow Appendix C.2.3. Arrows into the nonnumeric "not reached" band indicate that no run reached the corresponding effects, so no finite estimate is available.

and none for residual-BFM or Mixed-Gaussian LJS. The mixed-Gaussian ablation falls behind CEM despite retaining the same Jacobian coupling, showing that the optimizer also matters.

Completion does not capture motion quality. Figure 1 shows the distinction on shutter. CEM-LJS lifts the panel and moves through on its feet, whereas the best residual-BFM run reaches the same four acceptance conditions through a crossing with its torso on the floor. Direct latent search fails to establish the first effect and ends prone. Thus, the residual baseline's accepted effects do not imply comparable whole-body coordination. Door-cube comparisons are provided in Appendix B.

Search cost to reach a physical effect. Figure 3 includes the cost of unsuccessful attempts. On hurdle, residual-BFM reaches the first six effects at lower estimated cost, but only CEM-LJS completes the crossing and recovery. On shutter, residual-BFM reaches palm placement at lower cost, while CEM-LJS is more efficient from lifting onward. On door-cube, CEM-LJS has the lowest estimate at every effect. Inexpensive initial progress therefore does not necessarily translate into efficient task completion.

4.5 Ablations

We evaluate which components enable LJS to complete the four shutter effects. These seed-0 ablations use a total budget of 1,310,720 candidate control steps, separate from the per-effect decision caps in the main evaluation. BFM-Zero variants begin from the same simulator state and controller history and use the same effect definitions.

The action–latent coupling is load-bearing. CEM-LJS completes all four effects, while CEM direct-z completes none and CEM residual-BFM completes two. Keeping the optimizer fixed therefore exposes the importance of how proposals are represented and executed. Direct latent search retains coherent motion but misses the precise contacts required for acceptance. Residual search obtains partial progress through distorted postures and motions that do not support completion. Gaussian, MPPI, targeted refinement, and a random 29-dimensional subspace also fail in latent space: changing the optimizer or matching the proposal dimensionality does not recover the Jacobian coupling.

The optimizer matters within LJS. CEM-LJS completes the task using approximately half the candidate steps required by mixed-Gaussian LJS. Gaussian-only, MPPI, and LATTICE-style LJS complete no effects. Within the mixed-Gaussian variant, removing target-centered proposals yields zero effects, while removing zero-centered proposals yields two. Both populations are useful for that variant, but explicit targeted refinement is not necessary for CEM-LJS. Separately, direct latent Gaussian scales of 0.15, 0.5, and 1.5 all fail at hand placement under matched budgets, indicating that perturbation magnitude alone does not resolve the precision problem.

Table 2: **Shutter ablations, seed 0.** Search cost counts simulated candidate control steps; successful searches stop early. PPO counts training control steps and is not directly comparable. The SONIC entry excludes subsequent diagnostic searches.

Variant	Effects	Control steps
CEM-LJS	**4/4**	**696,320**
Mixed-Gaussian LJS	4/4	1,310,720
Zero-mean refinement LJS	0/4	1,310,720
Targeted-mean refinement LJS	2/4	1,310,720
Gaussian-only LJS	0/4	1,310,720
MPPI-LJS	0/4	1,310,720
LATTICE-style LJS	0/4	1,310,720
Gaussian direct-z	0/4	1,310,720
CEM direct-z	0/4	1,310,720
MPPI direct-z	0/4	1,310,720
Targeted refinement direct-z	0/4	1,310,720
Random 29-D latent subspace	0/4	1,310,720
CEM residual-BFM	2/4	1,310,720
PPO direct-z	0/4	1,310,720 training
PPO through LJS	0/4	1,310,720 training
Mixed-Gaussian LJS with SONIC	1/4	696,320

Using the score as an RL reward is insufficient at this budget. PPO trained with the negative physical-effect score completes no effects, either through direct latent commands or LJS-coupled proposals. The Jacobian mapping alone therefore does not make this objective easy to optimize through policy learning. These training interactions are distinct from candidate-rollout evaluations and do not establish a numerical search-efficiency comparison.

SONIC does not provide a substitute in this setting. Replacing BFM-Zero with SONIC (Luo et al., 2026) fails beyond the first effect. We test quantized FSQ tokens and continuous decoder inputs, mixed-Gaussian and CEM search, multiple noise scales and damping values, and longer horizons. None completes the interaction. The best recorded rollout raises the shutter to only $0.415\,\mathrm{m}$ against the required $0.75\,\mathrm{m}$, collapses into a low, unstable posture, and never crosses. Searches from pre-plateau states make no further progress. SONIC's ability to track supplied motions does not translate into discovering this interaction from physical-effect scores.

5 Conclusion

We presented Latent Jacobian Steer (LJS), a method for eliciting contact-rich OOD behaviors from frozen latent-conditioned humanoid controllers. LJS proposes physically meaningful changes in joint-action space, pulls them back into the latent space through a damped local inverse of the policy Jacobian, and evaluates the resulting latent commands through simulation. LJS can therefore be interpreted as an interface that makes behavioral foundation models *precise*. Importantly, LJS produces a sequence of latent commands $(z_1, \ldots, z_M)$ rather than an open-loop trajectory of joint actions. While each command is held fixed for a short execution window, the frozen policy continues to receive updated observations and generate new joint targets. These latent sequences can also serve as contact-rich demonstrations for behavioral cloning, reinforcement learning, or policy distillation; initialize subsequent task-specific optimization; and help probe which behaviors are present in a pretrained controller but difficult to access through direct prompting.

The current method is an offline behavior-generation procedure. It requires privileged simulator measurements, saved-state resets, and many parallel candidate rollouts, making it unsuitable for direct real-time deployment. Its success is also limited by the behavioral support of the frozen controller, while the decomposition into physical effects remains human-specified and can admit unintended solutions. A natural next step is to distill successful LJS-generated behaviors into closed-loop policies for efficient execution and eventual sim-to-real evaluation. We hope this simple and elegant method can give rise to new data-generation pipelines as well as learning paradigms that make use of pretrained controllers while not losing the authority of exploring in direct action space.

AI USE STATEMENT

We used generative AI tools to polish manuscript language, assist with literature discovery and summarization, and write and debug research code. AI assistance also supported the translation of author-provided natural-language task specifications into physical-effect measurements, scoring functions, and acceptance conditions, including iterative revisions informed by author feedback and analysis of simulation outputs. The research ideas, problem formulation, and overall research direction originated with the authors.

We reviewed all AI-assisted work, including manuscript text, code, physical-effect definitions, and interpretations of experimental results. The authors retained responsibility for methodological decisions and the claims reported in the paper. We take full responsibility for the final content of this work, including text, claims, and artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 describe the method, tasks, baselines, and evaluation protocol. Appendix A.2 specifies the CEM updates, Appendix C gives the physical-effect scores and acceptance conditions, and Appendix B provides qualitative rollout comparisons. All methods use the frozen BFM-Zero G1 controller and the same finalized task definitions. Each run used an NVIDIA GeForce RTX 5090 GPU on a standard workstation. We will release the code and benchmark scripts upon publication; the implementation records run configurations and decision-level traces to support verification. The videos for our runs are available at `https://drive.google.com/drive/folders/1bVWZjmiFkVYpr90T-_xrY52bcikuNSmC?usp=sharing`.

REFERENCES

Anurag Ajay, Aviral Kumar, Pulkit Agrawal, Sergey Levine, and Ofir Nachum. {OPAL}: Offline primitive discovery for accelerating offline reinforcement learning. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=V69LGwJ0lIN`.

Zixuan Chen, Mazeyu Ji, Xuxin Cheng, Xuanbin Peng, Xue Bin Peng, and Xiaolong Wang. Gmt: General motion tracking for humanoid whole-body control. *arXiv:2506.14770*, 2025.

Ziyang Cheng, Tianshu Tang, Jinxin Lan, Xinze Chen, Yuhan Gong, Zhichao Liu, Changzhong Wu, Yahao Mao, Zongyan Deng, Mingxuan Ma, Huasen Xi, Yilong Liu, Yutong Wu, Xiaofeng Wang, Borui Zhang, Bingyao Yu, Yang Wang, Yun Ye, Guan Huang, Xiaojie Jin, Zheng Zhu, and Jiwen Lu. Gigabrain-wbc-0.5: A behavior world model for robust humanoid whole-body tracking with environment interaction, 2026. URL `https://arxiv.org/abs/2608.18234`.

Alberto Silvio Chiappa, Alessandro Marin Vargas, Ann Zixiang Huang, and Alexander Mathis. Latent exploration for reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Prithwish Dan, Kushal Kedia, Angela Chao, Edward Duan, Maximus Adrian Pace, Wei-Chiu Ma, and Sanjiban Choudhury. X-sim: Cross-embodiment learning via real-to-sim-to-real. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 816–833. PMLR, 27–30 Sep 2025. URL `https://proceedings.mlr.press/v305/dan25a.html`.

Tony Feng, Trieu H. Trinh, Garrett Bingham, Dawsen Hwang, Yuri Chervonyi, Junehyuk Jung, Joonkyung Lee, Carlo Pagano, Sang hyun Kim, Federico Pasqualotto, Sergei Gukov, Jonathan N. Lee, Junsu Kim, Kaiying Hou, Golnaz Ghiasi, Yi Tay, YaGuang Li, Chenkai Kuang, Yuan Liu, Hanzhao Lin, Evan Zheran Liu, Nigamaa Nayakanti, Xiaomeng Yang, Heng-Tze Cheng, Demis Hassabis, Koray Kavukcuoglu, Quoc V. Le, and Thang Luong. Towards autonomous mathematics research, 2026. URL `https://arxiv.org/abs/2602.10177`.

Nikolaus Hansen, Anne Auger, Dimo Brockhoff, Dejan Tusar, and Tea Tušar. COCO: Performance Assessment. ArXiv e-prints, arXiv:1605.03560, May 2016. URL `https://inria.hal.science/hal-01315318`.

Jiajun Hu, Núria Armengol Urpí, Jin Cheng, and Stelian Coros. Maximum entropy behavior exploration for sim2real zero-shot reinforcement learning. In *Third Reinforcement Learning Beyond Rewards Workshop at the Reinforcement Learning Conference*, 2026. URL https://openreview.net/forum?id=2sjfMegaEb.

Zhi Jing, Siyuan Yang, Jicong Ao, Ting Xiao, Yu-Gang Jiang, and Chenjia Bai. Humanoidgen: Data generation for bimanual dexterous manipulation via LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Mk9ykil8eP.

Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6023–6029, 2019. doi: 10.1109/ICRA.2019.8794127.

Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE J. Robotics Autom.*, 3:43–53, 1987. URL https://api.semanticscholar.org/CorpusID:7687240.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Vince Kurtz. Hydrax: Sampling-based model predictive control on gpu with jax and mujoco mjx, 2024. https://github.com/vincekurtz/hydrax.

Joel Lehman, Jay Chen, Jeff Clune, and Kenneth O. Stanley. Safe mutations for deep and recurrent neural networks through output gradients, 2018. URL https://arxiv.org/abs/1712.06563.

Hongyi Li, Li Peizhuo, Yucheng Tao, Ze Wang, Fangzhou Xu, Jinyi Chen, Yanyan Yuan, Dapeng Jia, Yongbin Jin, Mingfeng Fan, Guillaume Sartoretti, and Hongtao Wang. Pgmt: Perceptive general motion tracking for humanoid robots, 2026a. URL https://arxiv.org/abs/2609.08511.

Jiaman Li, Jiajun Wu, and C Karen Liu. Object motion guided human motion synthesis. *ACM Trans. Graph.*, 42(6), 2023.

Jiaman Li, Alexander Clegg, Roozbeh Mottaghi, Jiajun Wu, Xavier Puig, and Karen Liu. *Controllable Human-Object Interaction Synthesis*, pp. 54–72. 11 2024. ISBN 978-3-031-72939-3. doi: 10.1007/978-3-031-72940-9_4.

Yitang Li, Zhengyi Luo, Tonghe Zhang, Cunxi Dai, Anssi Kanervisto, Andrea Tirinzoni, Haoyang Weng, Kris Kitani, Mateusz Guzek, Ahmed Touati, Alessandro Lazaric, Matteo Pirotta, and Guanya Shi. BFM-zero: A promptable behavioral foundation model for humanoid control using unsupervised reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=jkhl2oI0g5.

Qiayuan Liao, Takara E. Truong, Xiaoyu Huang, Yuman Gao, Guy Tevet, Koushil Sreenath, and C. Karen Liu. Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion. *Science Robotics*, 11(117):eadx8924, 2026. doi: 10.1126/scirobotics.adx8924. URL https://www.science.org/doi/abs/10.1126/scirobotics.adx8924.

Junjie Lu, Xinyao Qin, Yuhua Jiang, Kaixin Wang, Chuheng Zhang, Bin Liang, Jun Yang, Min Xu, and Li Zhao. Unisteer: Unified noise steering for efficient human-guided vla adaptation, 2026. URL https://arxiv.org/abs/2605.10821.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Fernando Castañeda, Sirui Chen, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, Jinhyung Park, David Sami, Zi Wang, Xingye Da, Runyu Ding, Cyrus Hogg, Lina Song, Edy Lim, Eugene Jeong, Tairan He, Haoru Xue, Wenli Xiao, Simon Yuen, Jan Kautz, Yan Chang, Umar Iqbal, Linxi "Jim" Fan, and Yuke Zhu. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *Science Robotics*, 11(117): eaed4592, 2026. doi: 10.1126/scirobotics.aed4592. URL https://www.science.org/doi/abs/10.1126/scirobotics.aed4592.

Tongzhou Mu, Minghua Liu, and Hao Su. Drs: Learning reusable dense rewards for multi-stage tasks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=6CZ50WgfCG.

Michael Murray, Daphne Chen, Simran Bagaria, Dean Fortier, Tess Hellebrekers, Galen Mullins, Harshavardhan Gajarla, Oier Mees, Maya Cakmak, and Andrey Kolobov. FlowDAgger: Human-in-the-loop adaptation of generative robot policies in latent space. In *Proceedings of the 10th Conference on Robot Learning*, 2026.

Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation policies with hilbert representations. In *Forty-first International Conference on Machine Learning*, 2024a. URL https://openreview.net/forum?id=LhNsSaAKub.

Seohong Park, Oleh Rybkin, and Sergey Levine. METRA: Scalable unsupervised RL with metric-aware abstraction. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=c5pwL0Soay.

Karl Pertsch, Youngwoon Lee, and Joseph J. Lim. Accelerating reinforcement learning with learned skill priors. In *Conference on Robot Learning (CoRL)*, 2020.

Cristina Pinneri, Shambhuraj Sawant, Sebastian Blaes, Jan Achterhold, Joerg Stueckler, Michal Rolinek, and Georg Martius. Sample-efficient cross-entropy method for real-time planning. In *Conference on Robot Learning 2020*, 2020. URL https://corlconf.github.io/paper_217.

Michael Posa, Cecilia Cantu, and Russ Tedrake. A Direct Method for Trajectory Optimization of Rigid Bodies Through Contact. *International Journal of Robotics Research (IJRR)*, 33(1):69–81, January 2014. doi: 10.1177/0278364913506757. URL http://ijr.sagepub.com/content/33/1/69.short.

Antonin Raffin, Jens Kober, and Freek Stulp. Smooth exploration for robotic reinforcement learning. In Aleksandra Faust, David Hsu, and Gerhard Neumann (eds.), *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1634–1644. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/raffin22a.html.

Nemanja Rakicevic, Antoine Cully, and Petar Kormushev. Policy manifold search: exploring the manifold hypothesis for diversity-based neuroevolution. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '21, pp. 901–909. ACM, 2021. doi: 10.1145/3449639.3459320. URL http://dx.doi.org/10.1145/3449639.3459320.

Nadun Ranawaka, Josiah Wong, Wei-Lin Pai, Wei-Teng Chu, Tianyuan Dai, Masoud Moghani, Hang Yin, Yunfan Jiang, Wesley Durbano, Brandon Huynh, Yu Fang, Danfei Xu, Ruohan Zhang, Li Fei-Fei, Linxi Fan, Bowen Wen, Ajay Mandlekar, and Yuke Zhu. Simfoundry: Modular and automated scene generation for policy learning and evaluation, 2026. URL https://arxiv.org/abs/2606.28276.

Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1(2):127–190, 1999. doi: 10.1023/A:1010091220143.

Pierre Schumacher, Daniel F.B. Haeufle, Dieter Büchler, Syn Schmitt, and Georg Martius. Dep-rl: Embodied exploration for reinforcement learning in overactuated and musculoskeletal systems. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*, May 2023. URL https://openreview.net/forum?id=C-xa_D3oTj6.

Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation. 07 2024. doi: 10.15607/RSS.2024.XX.061.

Lucy Xiaoyang Shi, Joseph J Lim, and Youngwoon Lee. Skill-based model-based reinforcement learning. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 2262–2272. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/shi23a.html.

Andy Tang, William Chen, Andrew Wagenmaker, Chelsea Finn, and Sergey Levine. Improving robotic generalist policies via flow reversal steering. In *3rd Workshop on Semantic Reasoning and Goal Understanding in Robotics (RSS 2026)*, 2026. URL https://openreview.net/forum?id=lZLHs6P90d.

Ilyass Taouil, Michal Ciebelski, Shafeef Omar, Haizhou Zhao, Angela Dai, Aaron M. Johnson, and Majid Khadiv. MotionDisco: Motion discovery for extreme humanoid loco-manipulation, 2026. URL https://arxiv.org/abs/2606.06139.

Andrea Tirinzoni, Ahmed Touati, Jesse Farebrother, Mateusz Guzek, Anssi Kanervisto, Yingchen Xu, Alessandro Lazaric, and Matteo Pirotta. Zero-shot whole-body humanoid control via behavioral foundation models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=9sOR0nYLtz.

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=2a96Bf7Qdrg.

Marc Toussaint. Logic-geometric programming: An optimization-based approach to combined task and motion planning. In *International Joint Conference on Artificial Intelligence*, 2015. URL https://api.semanticscholar.org/CorpusID:11309330.

Charles W. Wampler. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1): 93–101, 1986. doi: 10.1109/TSMC.1986.289285.

Max Wilcoxson, Qiyang Li, Kevin Frans, and Sergey Levine. Leveraging skills from unlabeled prior data for efficient online exploration. In *International Conference on Machine Learning (ICML)*, 2025.

Grady Williams, Andrew Aldrich, and Evangelos Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40:344–357, 01 2017. doi: 10.2514/1.G001921.

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, 2025. doi: 10.1109/ICRA55743.2025.11127320.

Runyi Yu, Xiaoyi Lin, Ji Ma, Yinhuai Wang, Koukou Luo, Jiahao Ji, Huayi Wang, Wenjia Wang, Runhan Zhang, Ping Tan, Ting Wu, Ruoli Dai, Qifeng Chen, and Lei Han. Omnicontact: Chaining meta-skills via contact flow for generalizable humanoid loco-manipulation, 2026. URL https://arxiv.org/abs/2606.26201.

Yanjie Ze, Zixuan Chen, Joao Pedro Araujo, Zi-ang Cao, Xue Bin Peng, Jiajun Wu, and Karen Liu. Twist: Teleoperated whole-body imitation system. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 2143–2154. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/ze25a.html.

A THEORY

A.1 FORWARD-BACKWARD REPRESENTATIONS AND BFM-ZERO

We briefly describe the forward-backward (FB) representation underlying BFM-Zero (Li et al., 2026b). For simplicity, we write the expressions in terms of the state s, suppressing the distinction between privileged state and observation history.

The FB representation learns a forward mapping F and a backward mapping B that together approximate the discounted future-state occupancy under the latent-conditioned policy π_z:

$$M^{\pi_z}(s, u, \mathrm{d}s') \approx F(s, u, z)^\top B(s')\, \rho(\mathrm{d}s'),$$

where ρ is the reference state distribution and

$$M^{\pi_z}(s, u, X) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbf{1}\{s_{t+1} \in X\} \,\middle|\, s_0 = s,\ u_0 = u,\ \pi_z\right].$$

Thus, F and B describe which future states a latent-conditioned policy is expected to visit.

For a state-based reward r, using the convention that rewards are evaluated at the next state, this factorization gives

$$Q_r^{\pi_z}(s, u) \approx F(s, u, z)^\top \mathbb{E}_{s' \sim \rho}[B(s')r(s')]\,.$$

The policy is trained to favor actions with large $F(s, u, z)^\top z$. Consequently, the reward-weighted backward features provide a latent command for a downstream reward:

$$z_r = \mathbb{E}_{s \sim \rho}[B(s)r(s)]\,.$$

This provides a way to prompt the pretrained policy without task-specific retraining. Similarly, a desired state can be embedded through B, while a reference motion can be represented by a sequence of state embeddings. The resulting commands are normalized to the controller's latent domain before execution.

These prompting mechanisms explain the behavioral structure of the pretrained controller. LJS instead searches latent commands through the frozen actor's Jacobian and simulator-evaluated physical effects; it does not use F or B during search.

A.2 CEM UPDATES FOR JOINT-ACTION SEARCH

We adapt the CEM implementation in Hydrax (Kurtz, 2024) to search over joint-action perturbations. At each planning step, we initialize

$$\mu_0 = 0, \qquad s_0 = \sigma \mathbf{1}_{29},$$

where $\sigma > 0$ is the initial exploration scale. We divide the candidate budget across R rounds. In round q, we draw N_q proposals and map them into latent commands:

$$\Delta u_i^{(q)} \sim \mathcal{N}\big(\mu_q, \mathrm{diag}(s_q^2)\big)\,, \qquad z_i^{(q)} = \Pi\Big(z + A\Delta u_i^{(q)}\Big)\,.$$

One proposal is set to the current mean, $\Delta u_1^{(q)} = \mu_q$, so the first round includes the unchanged command. The latent anchor z and pull-back map A remain fixed throughout these rounds.

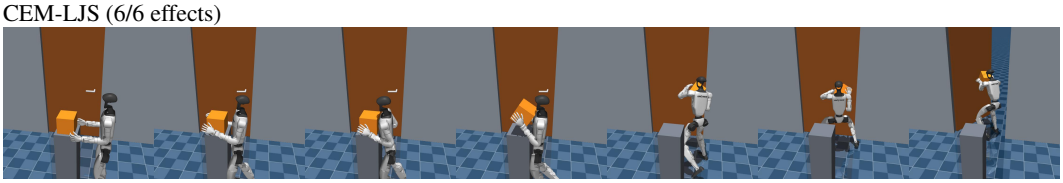
We simulate each candidate and rank admissible proposals by their physical-effect scores. Let $\mathcal{V}_q$ denote the admissible candidates. The elite set $\mathcal{E}_q$ contains the lowest-scoring

$$n_q^{\mathrm{elite}} = \min(|\mathcal{V}_q|, \max(2, \lceil \kappa N_q \rceil))$$

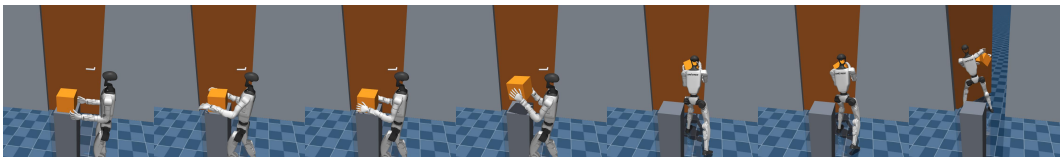
proposals, where $\kappa \in (0, 1]$ is the elite fraction. When the elite set is nonempty, we update

$$\mu_{q+1} = \frac{1}{|\mathcal{E}_q|} \sum_{i \in \mathcal{E}_q} \Delta u_i^{(q)},$$

CEM-LJS (6/6 effects)



CEM direct-z (6/6 effects)



CEM residual-BFM (4/6 effects)

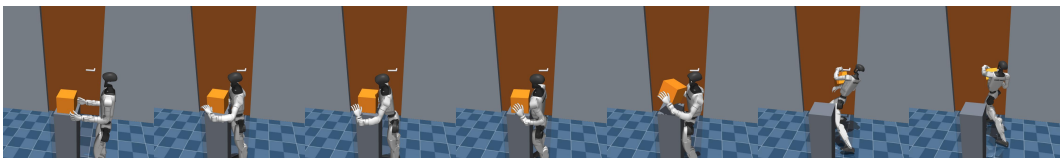


Figure 4: **Door opening while carrying a cube.** Seven chronological frames per method, showing pickup, carrying, and interaction with the door. CEM-LJS and the best direct-z run complete all six effects; the best residual-BFM run completes four. All rows use the same camera. Frames are selected around effect transitions and are not synchronized across methods.

$$s_{q+1,j} = \max\left(s_{\min}, \sqrt{\frac{1}{|\mathcal{E}_q|}\sum_{i\in\mathcal{E}_q}\left(\Delta u_{i,j}^{(q)} - \mu_{q+1,j}\right)^2}\right), \qquad j = 1,\ldots,29.$$

The floor $s_{\min} > 0$ prevents the proposal distribution from collapsing completely. If no candidate is admissible, we retain the previous distribution:

$$\mu_{q+1} = \mu_q, \qquad s_{q+1} = s_q.$$

After all rounds, we select the lowest-scoring admissible candidate evaluated across the entire planning step. Its latent command is executed for a short window before recomputing the Jacobian and restarting CEM from the resulting state. If no admissible candidate was found, the search terminates without declaring task completion.

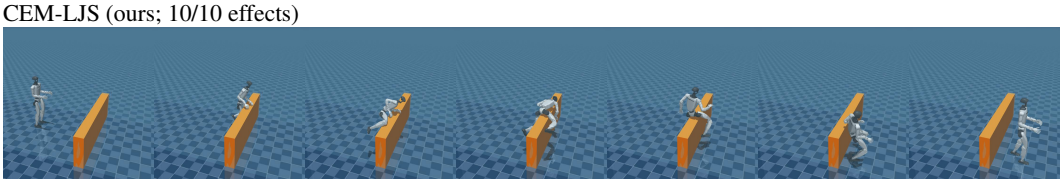
B Qualitative rollout comparisons

Figure 4 compares CEM-LJS with the best direct-latent and residual runs on door-cube. Both CEM-LJS and direct-z complete the six effects in the illustrated runs. Residual-BFM reaches carrying but fails to complete handle actuation and door opening. These examples show the resulting interactions; the aggregate completion and search-cost results are reported in the main paper.

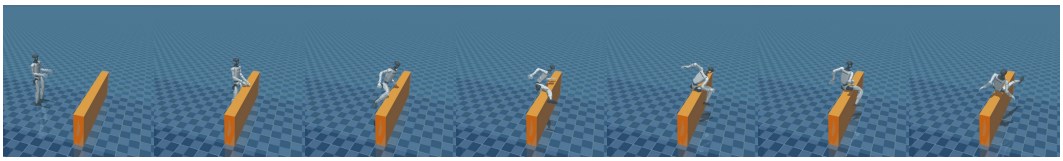
C Physical-effect definitions and human–AI score development

This appendix reports how the physical-effect definitions were developed and provides the mathematical scores and acceptance conditions for the three evaluated tasks. We would like to reiterate that our main contribution is LJS itself which makes the search for a specific physical effect productive. The following score functions are not part of our proposed LJS algorithm. They are to be viewed as evaluations of the kind of exploration that is possible with LJS, while impossible or highly sample-inefficient with other baselines. We leave the autonomous and efficient discovery of these physical effects for a given task in the broader sense to future work. The interaction card below follows the compact alternating-contribution format of Feng et al. (2026). The human specified the intended physical outcome, supplied the simulator conventions, corrected geometric and temporal

CEM-LJS (ours; 10/10 effects)



CEM direct-z (7/10 effects)



CEM residual-BFM (7/10 effects)

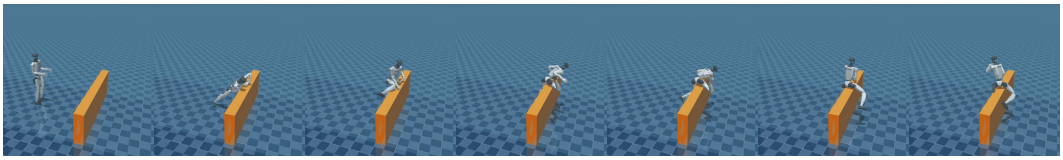


Figure 5: **Hurdle traversal.** Seven chronological frames from a successful CEM-LJS run and the best baseline runs. CEM-LJS crosses the hurdle and returns to standing. Direct-z and residual-BFM each accept seven effects but fail to clear the far edge with the left leg. All rows use the same fixed camera; frames are not synchronized across methods.

assumptions, and accepted the final definition. The language model translated those instructions into candidate measurement and score code. It did not observe rollouts, choose candidates, control the robot, or modify an effect during a reported search.

C.1 Human–AI interaction card: first hurdle effect

Table 3: **Human–AI interaction card for the first hurdle effect.** The card records the substantive revisions rather than reproducing the full prompts and code responses. The final scores and acceptance conditions are given in Appendix C.2.3.

Contributor	Contribution
Human	Established the coding protocol: define one physical effect at a time for LJS with frozen BFM-Zero; implement measurements, progress scores, and acceptance conditions using the supplied MuJoCo interface; use simple scaled residuals; keep score and acceptance separate; and do not introduce joint-pose commands, inverse kinematics, motion tracking, fine-tuning, teleportation, or external forces. Any revised score must be rerun from the preceding accepted effect.
Gemini 3.8 Flash	Restated the protocol and requested the task geometry, entity identifiers, and first-stage instruction before generating code.
Human	Specified the fixed hurdle, crossing and lateral axes, right-leg lead, palm-normal conventions, collision-surface geometry, and contact-force convention. Requested the first effect: place both palms on the hurdle top, facing downward, with separate bilateral support, while keeping the robot within the usable width. Either hand could settle anywhere suitable; no exact hand target or extra foot-lifting stage was allowed.
Gemini 3.8 Flash	Proposed a complete first-stage module using collision geometry for palm placement, palm-normal alignment, per-hand contact loads, and pelvis centering, together with initial model-chosen tolerances.

Table 3 continued.

Contributor	Contribution
Human	Corrected three implementation details: use each transformed hand geometry's bounding-box center for horizontal placement and its minimum vertex for height; measure support only at the final frame rather than claiming a sustained five-frame hold; and evaluate all surface bounds in the hurdle's live local frame instead of hardcoding its world center.
Gemini 3.8 Flash	Revised the measurements to use those distinct geometric statistics, instantaneous final-frame forces, and the hurdle transform recorded in each frame.
Human	Refined the intended outcome after inspecting the definition: keep both hands near the middle to leave room for the next leg movement, but avoid exact positions; require only reasonably downward palms and light bilateral load; reject side-friction support; keep the robot roughly centered; and stop penalizing small placement errors once the palms are suitably on top. Valid near-edge top contact had to count, so top support was to be identified from contact position and normal direction rather than an arbitrary edge strip.
Gemini 3.8 Flash	Added placement and orientation deadbands, a centered admissible band, light per-hand support targets, and top-face contact filtering using both local contact position and normal alignment. The human reviewed this proposal before the definition was incorporated into the finalized hurdle module.

Raw prompts and outputs: `https://share.gemini.google/GgqoaJR2Yw4F`

The card documents one representative effect rather than presenting the coding agent as an autonomous task designer. The successive corrections are part of the task-specification process: they remove assumptions that were inconsistent with the simulator semantics or with the intended physical outcome. The final reported searches use fixed, human-reviewed definitions.

C.2 Mathematical physical-effect definitions

We give the scores and acceptance conditions for the ten hurdle effects, four shutter effects, and six door-cube effects. Distances, forces, and times are in metres, newtons, and seconds. Angles are in radians unless degrees are shown. Measurements are taken at the final frame of the effect history ξ, except where a temporal minimum or maximum is specified.

Residual and acceptance conventions. Define elementwise residuals

$$L_a(v;b) = \frac{[b-v]_+}{a}, \qquad U_a(v;b) = \frac{[v-b]_+}{a}, \qquad N_{a,d}(v;b) = \frac{[|v-b|-d]_+}{a}.$$

Vector scales and tolerances are applied componentwise. For each effect we list residual blocks $r_{k,j}$; its score is

$$c_k(\xi) = \sum_j \|r_{k,j}(\xi)\|_2^2.$$

By default, acceptance requires every residual block to vanish. A superscript g marks a guidance-only block excluded from acceptance. Explicit acceptance overrides below replace the zero-residual test for the named block. Requirements marked c additionally reject a candidate when their acceptance test fails on the full predicted effect segment. The scores are evaluated over successive trajectory prefixes and averaged over the latter half of the prediction horizon, as in the main method.

C.2.1 Shutter traversal

Let $\mathcal{P}_G(s)$ be the transformed collision vertices of body group G, with $G = U$ for the head and torso and $G = R$ for the entire robot. The shutter's lower edge and far face are $y_b(s)$ and $x_f(s)$. Define

$$d_G(s) = \min_{p \in \mathcal{P}_G(s)} p_x - x_f(s), \qquad a_U(s) = \left[\max_{\substack{p \in \mathcal{P}_U(s) \\ p_x \le x_f(s)+.02}} p_z + .02 - y_b(s) \right]_+ ,$$

where $a_U = 0$ if the restricted vertex set is empty. For each hand, its placement point uses the mean vertex coordinates in x, y and the maximum vertex height. Let $e_h \in \mathbb{R}^6$ stack its displacement from the two targets $(x_{\text{panel}}, y_{\text{panel}} \pm .20, y_b - .005)$. Let $n_h \in \mathbb{R}^2$ contain the upward components of the palm normals. Finally, F_a is the signed net upward force on the panel from hand, wrist, and elbow contacts.

1. **Palms beneath the shutter.**
$$r_1 = \operatorname{col}\big(N_{.10,.04}(e_h; 0),\ L_1(n_h; \cos 30^\circ)\big).$$
Acceptance instead permits $\|e_h\|_\infty \le .10$ and both upward components at least $\cos 45^\circ$.
2. **Lift the shutter.**
$$r_2 = \operatorname{col}\big(L_{.10}(y_b; .75),\ L_{10}(F_a; 1)\big).$$
Acceptance requires a lower-edge height of at least .75 and at least 1 N of upward arm contact.
3. **Head and torso through.**
$$r_3 = \operatorname{col}\big(U_{.10}(a_U; 0),\ L_{.10}(d_U; .02)\big).$$
4. **Whole body clear.**
$$r_4 = L_{.10}(d_R; .02).$$

The evaluated shutter task ends at whole-body clearance. These four effects do not impose a standing or floor-contact condition after passage.

C.2.2 Door opening while carrying a cube

Let p_c, R_c, a denote the cube centre, rotation, and half-extents. The palm contact point is the collision-mesh mean shifted to the inward-most palm surface along the hand's local inward axis. For sides $\epsilon \in \{+1, -1\}$, define stacked palm errors relative to cube-local targets

$$e_{\text{reach},\epsilon} = p_{h,\epsilon} - p_c - R_c(a_x + .03, \epsilon(a_y + .02), 0)^\top, \qquad e_{\text{grasp},\epsilon} = p_{h,\epsilon} - p_c - R_c(.045, \epsilon a_y, 0)^\top.$$

Let $v_f \in \mathbb{R}^2$ be the dot products between normalized elbow-to-palm directions and the cube's forward axis. Let $b = (R_c^\top(p_c - p_{\text{torso}}))_{xy}$, $u = (R_{\text{torso}})_{zz}$, and v be the maximum root linear speed over the last .10 s. The maximum cube translation and rotation relative to its pose at the beginning of the effect are

$$m_p = \max_{s \in \xi} \|p_c(s) - p_c(s_0)\|_2, \qquad m_R = \max_{s \in \xi} \arccos\left(\frac{\operatorname{tr}(R_c(s_0)^\top R_c(s)) - 1}{2}\right),$$

with the arccos argument clipped to $[-1, 1]$. Define

$$r_m = U_{(.06, 15^\circ)}\big((m_p, m_R); (.02, 5^\circ)\big), \qquad h_c = p_{c,z} - \sum_j |R_{c,zj}| a_j, \qquad h_s = h_c - z_{\text{stand}}.$$

Here z_{stand} is the stand's top height. F_s is the sum of positive cube–stand normal loads. A bar below a measurement denotes its minimum over the final .08 s; a bar above denotes its maximum over that window.

Contact and separation measurements. $F_h \in \mathbb{R}^2$ contains the minimum inward palm loads over .08 s. For hand ϵ, a contact contributes when its cube-local position q satisfies $|q_y - \epsilon a_y| \le .02$, $|q_x| \le .075$, and $|q_z| \le .075$. Its contribution is $[-\epsilon(R_c^\top f_c)_y]_+$, where f_c is force on the cube. F_h^{full} instead permits $|q_x| \le a_x + .01$ and $|q_z| \le a_z + .01$. These load measurements return zero if the history is shorter than .08 s.

Let p_H, e_H, l_H, r_H denote the handle centre, axial direction, half-length, and radius. Its sampled surface gap to the cube is

$$g_H = \left[\min_{t \in \{-1, -.5, 0, .5, 1\}} \Big\| \big[\big|R_c^\top(p_H + t l_H e_H - p_c)\big| - a\big]_+ \Big\|_2 - r_H\right]_+.$$

For door centre p_D, normal n_D, and half-thickness a_D, define

$$g_D = \left[|(p_c - p_D)^\top n_D| - \sum_j |(R_c^\top n_D)_j| a_j - a_D \right]_+ .$$

$F_H^\downarrow$ is the minimum summed downward cube force on the handle over .08 s; F_D is the minimum summed positive cube–panel normal load over the same window. Both return zero for shorter histories. The handle and door angles are θ_H and θ_D.

1. **Approach the cube.**
$$r_1 = \mathrm{col}\big(N_{.20,.04}(e_{\mathrm{reach}}; 0),\ L_{.4}(v_f; \cos 30^\circ),\ N_{(.18,.20),(.04,.10)}(b; (.25, 0)),\ r_m^{\mathrm{c}},\ U_{.5}(v; .25),\ L_{.3}(u; \cos 25^\circ)\big).$$
The motion block accepts $m_p \le .03$ and $m_R \le 8^\circ$; these are also candidate constraints.
2. **Grasp opposing side faces.**
$$r_2 = \mathrm{col}\big(N_{.15,.02}(e_{\mathrm{grasp}}; 0),\ L_5(F_h; 5),\ r_m^{\mathrm{c}},\ U_{.5}(v; .25)\big).$$
Each palm need only sustain 3 N for acceptance. The motion override is unchanged.
3. **Lift clear of the stand.**
$$r_3 = \mathrm{col}\big(L_{.12}(h_s; .08),\ L_5(F_h; 5),\ U_5(F_s; 0),\ L_{.3}(u; \cos 30^\circ)\big).$$
Acceptance permits $F_s \le .1$ and requires each palm load to be at least 3 N.
4. **Carry to the handle.**
$$r_4 = \mathrm{col}\big(U_{.20}(g_H; .02),\ (\sqrt{.5}\, N_{.20,.04}(e_{\mathrm{grasp}}; 0))^{\mathrm{g}},\ L_{.20}(\underline{h}_c; .55)^{\mathrm{c}},\ L_5(F_h; 5),\ U_5(\overline{F}_s; 0)^{\mathrm{c}},\ L_{.3}(u; \cos 30^\circ)\big).$$
The palm-load override remains 3 N per hand; the stand-load override and constraint permit $\overline{F}_s \le .1$.
5. **Depress the handle with the cube.**
$$r_5 = \mathrm{col}\big(L_{.35}(\theta_H; .50),\ L_8(F_H^\downarrow; 8),\ U_{.10}(g_H; 0),\ U_{.15}(\theta_D; .04),\ L_{.20}(h_c; .15),\ L_5(F_h; 5)^{\mathrm{c}},\ L_{.3}(u; \cos 30^\circ)\big).$$
Acceptance permits $\theta_D \le .06$ and requires $F_H^\downarrow \ge 3$ and both palm loads at least 3 N. The palm-load condition is also a candidate constraint.
6. **Push the door open.**
$$r_6 = \mathrm{col}\big(L_{.5}(\theta_D; 1),\ L_5(F_D; 5),\ U_{.10}(g_D; 0),\ L_{.20}(h_c; .15),\ L_5(F_h^{\mathrm{full}}; 5),\ L_{.3}(u; \cos 30^\circ)\big).$$
Acceptance requires $\theta_D \ge .95$, $F_D \ge 3$, and each full-face palm load at least 3 N.

C.2.3 HURDLE TRAVERSAL

The fixed hurdle has near and far faces x_n, x_f, top height z_h, and half-width w. The right leg leads and the left follows. For collision group G, let x_G, z_G be the minimum world x, z coordinates of its collision surface. Mesh vertices and sphere extrema provide these measurements. Write T_G and F_G for its upward hurdle-top load and positive floor normal load. A top contact is counted only if its height is at least $z_h - .015$ and its contact normal has vertical magnitude greater than .7. Let A_n be the summed positive normal load on hands, wrists, and elbows at contacts with $x \le x_n + .015$ and $z < z_h - .015$. Define

$$C_b = U_{.20}(|y_{\mathrm{root}}|; b), \qquad K = U_{20}(A_n; 2), \qquad B(\xi) = \left[\cos 35^\circ - \min_{s \in \xi} (R_{\mathrm{torso}}(s))_{zz} \right]_+ .$$

K is always guidance-only. Let $u = (R_{\mathrm{torso}})_{zz}$, $l = (R_{\mathrm{torso}})_{xz}$, $f_y = (R_{\mathrm{torso}})_{yx}$, and W be total model mass times gravitational acceleration. Subscripts R, L denote right and left feet; $\mathrm{leg}R, \mathrm{leg}L$ include collision bodies descending from the respective knee. $A, P, \mathcal{R}$ denote arms, pelvis, and entire robot.

Placement and clearance measurements. For each palm's collision bounding-box centre b_h, define

$$e_h = \operatorname{col}\left(b_{h,x} - \operatorname{clip}(b_{h,x}, x_n + .025, x_f - .025),\ b_{h,y} - \operatorname{clip}(b_{h,y}, -w_p, w_p),\ z_h^{\text{hand}} - z_h\right),$$

where $w_p = \min(.35, w - .08)$ and z_h^{hand} is the hand's lowest collision point. Stack both hands into e_H; n_H stacks downward palm-normal components. Let g_G be the minimum vertex distance of group G to the top rectangle $[x_n + .02, x_f - .02] \times [-w + .08, w - .08] \times \{z_h\}$. The pelvis error e_P comprises its bounding-box centre's signed distance from $[x_n + .02, x_f - .02]$ in x and its lowest collision height minus z_h.

Define right-foot and trailing-leg clearance errors

$$a_R = \begin{cases} [z_h + .02 - z_R]_+, & x_R < x_f + .02, \\ 0, & \text{otherwise}, \end{cases} \qquad a_L = \left[z_h + .02 - \min_{\substack{p \in \mathcal{P}_{\text{leg}L} \\ p_x < x_f + .02}} p_z \right]_+,$$

with $a_L = 0$ when no vertices remain. Let $d_L = [.05 - \min(d_{\text{knee}}, d_{\text{ankle}})]_+$, where each d is the left joint's displacement from the pelvis projected onto the torso's forward axis. Finally, $v = \|\dot{p}_{\text{root}}\|_2$ and $\omega = \|\omega_{\text{root}}\|_2$.

1. **Both palms on top.**

$$r_1 = \operatorname{col}\big(C_{.36}, U_{.20}(|e_H|; .035), L_1(n_H; \cos 40^\circ), L_2(T_{H_L}; 2), L_2(T_{H_R}; 2)\big).$$

2. **Right leg on top for support.**

$$r_2 = \operatorname{col}\big(C_{.36}, U_{.20}(g_{\text{leg}R}; 0), L_{.20}((p_{\text{knee}R,x}, p_{\text{knee}R,z}); (x_n + .04, z_h + .08))^{\text{g}}, L_{10}(T_{\text{leg}R}; 10), L_{20}(\min_{s \in \xi} T_A(s); 5)^{\text{c}}, L_{.20W}(T_A; .20W)^{\text{g}}, K^{\text{g}}\big).$$

   The leg-gap block accepts $g_{\text{leg}R} \le .025$.

3. **Right foot on top.**

$$r_3 = \operatorname{col}\big(L_{.20}(x_R; x_n + .02), L_{.20}(z_R; z_h - .01), L_5(T_R; 5), L_{20}(F_L; 5), K^{\text{g}}\big).$$

4. **Lean and sweep the right foot across.**

$$r_4 = \operatorname{col}\big(C_{.36}, L_{.30}(l; \sin 30^\circ), U_{.20}(a_R; 0), L_{.20}(x_R; x_f + .02), L_{20}(F_L; 5), L_{10}(T_{\mathcal{R}}; 10), K^{\text{g}}\big).$$

5. **Lift the left foot into a supported straddle.**

$$r_5 = \operatorname{col}\big(C_{.35}, L_{.30}(l; \sin 10^\circ), L_{.20}(x_R; x_f + .02), L_{.20}(z_L; .08), U_W(F_L; 0)^{\text{g}}, L_{.20}(x_{\text{root}}; x_n + .02), L_{20}(T_{\mathcal{R}}; 10), K^{\text{g}}\big).$$

6. **Settle into an upright supported straddle.**

$$r_6 = \operatorname{col}\big(C_{.36}, L_{.30}(u; \cos 15^\circ), L_{.30}(f_y; \cos 30^\circ), e_P/.20, L_{.20}(x_R; x_f + .02), U_{.30}(p_{\text{ankle}R,z} - p_{P,z}; -.10), L_{.20}(z_L; .08), U_{.50}(v; .20), U_1(\omega; .50), L_{10}(T_P; 10), K^{\text{g}}\big).$$

   The pelvis-placement block accepts $\|e_P\|_\infty \le .025$.

7. **Left foot on the ledge.**

$$r_7 = \operatorname{col}\big(g_L/.20, d_L/.20, L_{20}(T_L; 10), L_{.20}(z_L; z_h - .01)^{\text{g}}, B(\xi)/.20\big).$$

   Acceptance requires $g_L/.20 \le .025$, $d_L = 0$, $T_L \ge 10$, and terminal $u \ge \cos 35^\circ$. The historical balance penalty may remain nonzero.

8. **Left leg clears the far edge.**

$$r_8 = \operatorname{col}\big(L_{.20}(\min_{s \in \xi} x_L(s); x_n - .04)^{\text{c}}, L_{.20}(x_L; x_f + .03), a_L/.20, B(\xi)/.20\big).$$

   The forward residual accepts $L_{.20}(x_L; x_f + .03) \le .05$; the remaining blocks use their default zero tests.

9. **Both feet on the far-side floor.**

$$r_9 = \mathrm{col}\big(L_{.20}(\min_{s\in\xi} x_L(s); x_f + .01)^{\mathrm{c}},$$
$$\{L_{.20}(x_i; x_f + .03), U_{.20}(z_i; .04), L_{10}(F_i; 5)\}_{i\in\{L,R\}}, (B(\xi)/.20)^{\mathrm{g}}\big).$$

10. **Stand and settle.** Let $p^\star = (x_f + .50, y_{\mathrm{root}}, z_{\mathrm{nom}})$ and let e_{rot} be the quaternion rotation difference from the identity orientation. With nominal joint positions q_{nom}, define

$$e_{\mathrm{stand}} = \mathrm{col}\big((p_{\mathrm{root}} - p^\star)/.20, e_{\mathrm{rot}}/.5, q_{\mathrm{joints}} - q_{\mathrm{nom}}\big), \qquad r_{10} = \mathrm{col}\big(e^{\mathrm{g}}_{\mathrm{stand}}, [\overline{v} - .15]_+\big).$$

Here temporal extrema use the final .5 s. Acceptance requires at least .5 s of history, $\underline{u} \geq \cos 15^\circ$, $\overline{v} \leq .15$, $\min_{i\in\{L,R\}} \underline{F}_i \geq 5$, and $\min_s(x_{\mathcal{R}}(s) - x_f) \geq .03$ over that window.

D TIPS FOR PRACTITIONERS

Begin with an intuitive score. Highly constrained scores are not necessarily better scores. We first encode the obvious physical outcome and inspect the behavior produced by the search. We add terms only to address failure modes observed in the resulting rollouts, rather than trying to anticipate every possible failure in advance.

Define the task one physical effect at a time. The terminal rollout of one accepted effect often reveals how the next effect should be specified. It may expose the relevant contact configuration, show which movement is currently blocked, or indicate that an intermediate physical effect is required. We therefore found it more practical to construct long tasks stage-wise than to define every effect in advance.

Inspect the motion, not only the scalar score. A lower physical-effect score does not always indicate useful progress, and a temporary score increase does not always indicate failure. The robot may need to make a short adjustment before further progress becomes possible, while a search can also exploit a missing condition and reduce the score through an unintended movement. Video replay remains the most useful diagnostic for distinguishing these cases.

Avoid ending a stage in uncontrolled motion. Dynamic or unexpected terminal states are difficult to recover from during later effects. When this occurs, the next score should not merely force the robot forward from an unsuitable state. It is usually better to revise the preceding effect to require settling, support, or bounded velocity before acceptance.

Match the horizon to the physical response. Large body movements and heavy-object displacement may not develop within the same horizon as a small contact adjustment. Increasing the prediction horizon gives the simulator time to reveal the outcome of a candidate, while increasing the execution horizon can reduce the number of replanning decisions. Both can also make an incorrect candidate more costly. We therefore treat prediction and execution horizons as task-stage parameters and report them for every effect.

E RESULTS BY SEED

Table 4 reports the runs underlying the main comparison. Each entry gives accepted effects and total simulated candidate control steps (millions), including unsuccessful attempts. Seed order is $(0, 1, 42)$ for hurdle and shutter, and $(1, 42, 123)$ for door-cube.

Table 4: **Individual evaluation runs.** Entries are accepted effects / candidate steps (10^6).

Task	Method	First seed	Second seed	Third seed
Hurdle	CEM-LJS	10 / 3.768	10 / 3.891	1 / 1.475
Hurdle	CEM direct-z	7 / 3.359	5 / 3.748	5 / 3.625
Hurdle	CEM residual-BFM	7 / 2.068	1 / 0.369	6 / 3.195
Hurdle	Mixed-Gaussian LJS	4 / 2.540	1 / 0.410	5 / 3.953
Shutter	CEM-LJS	4 / 0.696	4 / 0.778	4 / 0.901
Shutter	CEM direct-z	0 / 0.614	0 / 0.614	0 / 0.614
Shutter	CEM residual-BFM	1 / 0.922	3 / 1.679	4 / 1.516
Shutter	Mixed-Gaussian LJS	0 / 0.614	0 / 0.614	0 / 0.614
Door-cube	CEM-LJS	6 / 1.147	4 / 1.434	6 / 0.922
Door-cube	CEM direct-z	3 / 1.475	6 / 1.372	1 / 1.024
Door-cube	CEM residual-BFM	1 / 0.942	4 / 1.290	1 / 0.942
Door-cube	Mixed-Gaussian LJS	3 / 1.475	4 / 1.495	0 / 0.614