BGDL: LEARNING WHOLE-BODY HUMANOID CONTROL FROM DEMONSTRATIONS WITH PHYSICAL ACTION GEOMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

Demonstrations are useful for whole-body humanoid control, but their value is not uniform along a trajectory. In permissive states, many actions can still complete the task, whereas near foothold limits, balance boundaries, or required contacts, small deviations can change the outcome. We introduce **Boundary Guided Demonstration Learning (BGDL)**, which uses this variation in local action freedom to determine where demonstration guidance should be strongest. BGDL derives state-dependent guidance from physical margins, proximity to required interactions, and near-term failure risk, strengthening imitation where action choices are more restricted while leaving greater freedom for reinforcement learning elsewhere. We further characterize how allowable deviation from a demonstrated action depends on margin reserve, local action sensitivity, and margin-estimation error, giving a local sufficient condition for preserving a one-step physical-margin condition. We evaluate BGDL on five whole-body humanoid tasks: stair climbing, curb traversal, ground object pick-up, chair sitting, and object carrying, and transfer the learned policies to a Unitree G1 without policy fine-tuning. BGDL improves average success from 86.0% to 91.2% over reference tracking while reducing falls from 12.8% to 8.0%. With matched demonstration data, total imitation weight, and environment interactions, BGDL improves success by 3.4 percentage points over random assignment of the same guidance weights. This suggests that the placement of demonstration supervision matters beyond its total amount.

1 INTRODUCTION

A humanoid climbing a staircase must place its foot accurately and transfer support at the right moment, but it does not need to reproduce every joint configuration of a demonstrated motion. The same is true for whole body interaction. A robot may approach an object or a chair in several reasonable ways, while the final grasp, foothold, or seat contact leaves much less room for error. This means that a demonstration is not equally important at every point in a trajectory. Some states allow many successful actions, while others require the policy to stay much closer to the demonstrated behavior. Human motion has become an increasingly practical source of supervision for humanoid control. H2O and OmniH2O use retargeted human motion for whole body teleoperation and skill learning (He et al., 2024b;a). ExBody shows that expressive motion can be preserved without forcing the lower body to track every detail of the reference (Cheng et al., 2024). HumanPlus uses humanoid shadowing to collect whole body demonstrations for autonomous behavior (Fu et al., 2024). Other recent systems use motion imitation as a shared control representation or combine retargeted motion with reinforcement learning and behavior cloning (He et al., 2025b; Ze et al., 2025). AMO improves adaptive whole body control (Li et al., 2025a), while ASAP focuses on transferring agile skills across the simulation to real dynamics gap (He et al., 2025a).

There has also been substantial progress on contact rich tasks and on obtaining useful demonstrations for them. WoCoCo organizes long horizon behaviors around a sequence of contacts (Zhang et al., 2024), while guided reinforcement learning for multi contact loco manipulation uses a trajectory optimized demonstration to support policy learning (Sleiman et al., 2025). OKAMI retargets a single RGB-D human demonstration to new object configurations and uses the resulting executions to train closed loop manipulation policies (Li et al., 2025b). GRAIL takes this further by generating robot

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.

compatible humanoid demonstrations from specified 3D scenes and video priors (Xie et al., 2026). These methods have made it easier to obtain, transfer, and use demonstrations. Our question starts after that point. **Once a useful demonstration is available, how much should the policy be asked to follow it at each state?** This question is different from deciding whether demonstrations are useful at all. Existing methods already show that expert supervision need not be applied uniformly. Q-filtered behavior cloning keeps expert supervision when the estimated action value favors the demonstrated action (Nair et al., 2018). ThriftyDAgger uses novelty and estimated task risk to decide when expert intervention is needed (Hoque et al., 2022), and AIM learns an intervention rule from a proxy value function that relates agent and expert behavior (Cai et al., 2025). These methods choose supervision using value, novelty, risk, or agreement with the expert.

For whole body control, however, a different issue also matters. A state may need precise guidance even when it is not especially unsafe or low value. Making hand contact during a grasp or making seat contact while sitting are intended parts of the task, but small action changes near those moments can still determine success or failure. Away from these states, several actions may work equally well. Forcing the policy to track the demonstration closely there can limit its ability to adapt to a different scene or discover a better motion.

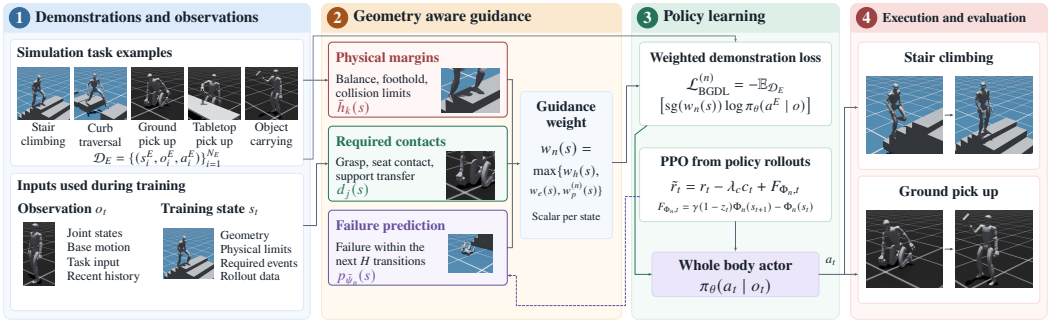

Figure 1: **Overview of Boundary Guided Demonstration Learning (BGDL).** BGDL uses physical margins, proximity to required interactions, and near-term failure risk to decide where demonstration guidance should be stronger. The resulting state-dependent weight scales the demonstration loss while reinforcement learning continues to optimize task behavior. Guidance increases near restrictive physical conditions and required interactions and decreases where the policy has more freedom to adapt.

We study this problem through the local action freedom around a demonstrated state. Physical constraints can be viewed as defining $\mathcal{A}_{\mathrm{phys}}(s) = \{a \in \mathcal{A} : h_k(f(s,a)) \geq 0,\ k = 1, \ldots, K\}$. Our approach does not try to recover this set directly. Instead, it uses physical margins, proximity to required interactions, and near-term failure information as signals of when the available action choices become more restricted. The idea is simple. When many actions remain acceptable, the policy should have room to adapt. When small deviations can violate a physical constraint or miss an important interaction, the demonstrated action should have more influence. We introduce *Boundary Guided Demonstration Learning* (BGDL), shown in Fig. 1, to apply this idea during reinforcement learning. BGDL builds a state-dependent demonstration weight from balance, foothold, and collision margins, distances to required interactions, and failure risk learned from policy rollouts. The weight increases the influence of demonstration actions where deviations are more consequential and reduces it where the policy has more freedom. Reinforcement learning still improves the task behavior, but the demonstration constrains that learning more strongly only where it is needed. **Our main contributions are**:

1. **A local action freedom view of demonstration learning.** We study demonstration supervision in terms of how much freedom the policy has around a demonstrated action. This separates states where many actions remain acceptable from states where balance, foothold, collision, or required interaction constraints leave much less room for deviation.
2. **Boundary Guided Demonstration Learning for whole body humanoid control.** BGDL uses physical margins, required interactions, and near term failure prediction to decide where demonstration guidance should be stronger. This allows reinforcement learning

to move beyond the demonstrations where several actions are acceptable while retaining stronger guidance near more restrictive states.

3. **Analysis and controlled evaluation of where demonstration guidance matters.** We relate allowable deviation from a demonstrated action to physical margin, local action sensitivity, and margin estimation error. We then test the role of guidance placement using controls matched for demonstration data, policy capacity, interaction budget, and total imitation weight, with evaluation across five whole body tasks in simulation and on a physical humanoid.

2 Related Work

Humanoid control from motion and demonstrations. Reference tracking and learned motion priors have enabled complex whole-body behaviors in simulation and on physical humanoids. DeepMimic combines motion tracking with task reward, while AMP replaces explicit trajectory tracking with an adversarial motion prior (Peng et al., 2018; 2021). More recent humanoid systems use retargeted motion, teleoperation, and behavior cloning for versatile whole-body control, including ExBody, HOVER, OmniH2O, HumanPlus, and TWIST (Cheng et al., 2024; He et al., 2025b; 2024a; Fu et al., 2024; Ze et al., 2025). GRAIL further generates robot-compatible loco-manipulation demonstrations from 3D assets and video priors (Xie et al., 2026). These works primarily address how demonstrations are obtained or converted into coordinated robot behavior. BGDL instead studies how strongly individual demonstration states should influence policy optimization, based on the local physical action geometry of the task.

Selective demonstration supervision. Prior work has already shown that expert supervision need not be applied uniformly. Q-filtered behavior cloning retains imitation when the critic prefers the demonstrated action to the policy action (Nair et al., 2018), while interactive imitation methods such as ThriftyDAgger and AIM decide when expert intervention is useful using novelty, risk, or agent–expert disagreement (Hoque et al., 2022; Cai et al., 2025). BGDL differs in the criterion used for selection. It uses physical margins, required-contact structure, and learned failure risk to identify states where the admissible action region becomes restrictive. This distinction is important for humanoids: a grasp or seat contact can require precise guidance without representing an unsafe state. We therefore compare BGDL against value-based and matched-weight selectors to isolate whether physical action geometry provides useful structure beyond generic risk or value estimates.

Contact structure and safe reinforcement learning. Contact-aware methods such as WoCoCo organize whole-body learning around sequential contact transitions (Zhang et al., 2024), while safe RL methods constrain policy optimization, trigger recovery, or filter actions using learned constraints or control barrier functions (Achiam et al., 2017; Thananjeyan et al., 2021; Yang et al., 2025). BGDL uses related physical information for a different purpose: it does not decompose the task into fixed contact stages or impose a runtime safety filter. Instead, physical and contact cues determine where stored demonstrations should exert stronger influence during policy learning.

3 Boundary Guided Demonstration Learning

BGDL trains a humanoid policy from its own experience and a fixed set of recorded demonstrations. Policy rollouts provide task rewards, physical costs, and failure outcomes. Demonstrations provide reference actions at observations recorded during an expert execution. The role of BGDL is to determine how much each reference action contributes to learning. Physical margins, required contact information, and predicted failure determine a scalar weight for each sample. The same actor learns from these weighted samples and from reinforcement learning throughout training.

3.1 Problem Setup

Let s_t denote the training state and o_t the observation available to the robot, including proprioception, task inputs, and observation history. The actor $\pi_\theta(a_t \mid o_t)$ produces a policy command a_t. Privileged simulator information can be used by the critic and the training cues, but is not added to the actor input. Demonstration actions must use the same coordinates as the policy likelihood. Any subsequent scaling, clipping, or mapping to joint targets belongs to the command interface and must be

distinguished from the action being scored. Training uses a fixed dataset $\mathcal{D}_E = \{(s_i^E, o_i^E, a_i^E)\}_{i=1}^{N_E}$ and a rollout batch $\mathcal{D}_n$ collected by the current policy at training cycle n. Here N_E counts stored samples, not complete trajectories. In a supervised update, the current actor is evaluated at o_i^E, the recorded action a_i^E is its target, and s_i^E supplies the information needed to weight that target. In a rollout update, the actor learns from the consequences of its own actions. These two sources update the same policy, but they enter through separate losses. Writing $\ell_{\mathrm{demo}}(\theta; o, a^E)$ for the demonstration loss and $\mathcal{J}_{\mathrm{RL}}$ for the rollout objective, the learning problem takes the form

$$\mathcal{J}_n(\theta) = \mathcal{J}_{\mathrm{RL}}(\theta) - \lambda_{\mathrm{IL}} \mathbb{E}_{(s,o,a^E)\sim\mathcal{D}_E} \left[w_n(s) \ell_{\mathrm{demo}}(\theta; o, a^E) \right]. \tag{1}$$

The coefficient λ_{IL} sets the overall strength of imitation. The weight $w_n(s) \in [0, 1]$ sets the relative influence of a particular sample and is held fixed during the actor update. We first define the information used to compute this weight, then specify the likelihood loss and the complete PPO objective.

3.2 Local Physical Action Geometry

Let $f(s, a)$ denote the state after one control interval, and let h_k be a physical margin that is nonnegative when constraint k is satisfied. The corresponding set of admissible actions is $\mathcal{A}_{\mathrm{phys}}(s) = \{a \in \mathcal{A} : h_k(f(s, a)) \geq 0,\ k = 1, \ldots, K\}$. This set describes satisfaction of the listed physical constraints after the transition. It does not by itself require progress toward a grasp or another task interaction. BGDL does not reconstruct the set or search its boundary. It uses physical margins, required event distances, and a learned failure estimate as cues for states where demonstration guidance may be important.

Physical margin. For constraint k, an estimate $\widehat{h}_k(s)$ is positive on the admissible side and negative when the estimated constraint is violated. The margins in our tasks concern balance, foothold clearance, and forbidden collisions. We normalize the margins using fixed positive scales c_k and retain the smallest normalized value, with $\bar{h}_k(s) = \widehat{h}_k(s)/c_k$ and $h_{\min}(s) = \min_k \bar{h}_k(s)$. The scales make the comparison meaningful across quantities with different units. A small $h_{\min}$ indicates proximity to at least one estimated physical limit. For example, a foot placed close to a tread edge has less foothold clearance than one placed farther inside the landing surface. This is a cue about physical clearance, not a direct measurement of how every action perturbation will affect the next state.

Required interaction. Some states require precise behavior even when the physical margins are positive. A hand must reach an appropriate contact region to establish a grasp, and the body must reach the seat during sitting. For an active event $j \in \mathcal{J}(s)$, let $d_j(s) \geq 0$ denote the normalized distance to its target. The active set follows the task phase, and the event features specify the relevant position, orientation, or relative motion. A zero distance means that the defined event target has been reached. It is not a physical violation, nor does it necessarily mean that the entire task is complete. Keeping this quantity separate from the physical margins allows a desired contact to receive guidance without being penalized as a collision.

Near-term failure. Physical margins and event distances can miss failures caused by the interaction of posture, momentum, and contact. We therefore fit a predictor $p_{\bar{\psi}_n}(s)$ from previous rollout outcomes to estimate failure within the next H transitions under recent collection policies. Unlike the physical and event definitions, this cue is learned. The parameters $\bar{\psi}_n$ identify the predictor held fixed during cycle n. A successful expert continuation is not itself a label of low risk under the current policy. The outcome labels, treatment of incomplete windows, and fitting and calibration splits are described in Appendix D.

3.3 Geometry Conditioned Demonstration Supervision

The three cues have different meanings and units. We convert each to a weight in $[0, 1]$ before determining a sample's contribution to the demonstration loss. Let $\sigma(x) = (1 + e^{-x})^{-1}$.

Physical margin guidance. We map the minimum normalized margin to

$$w_h(s) = \sigma\left(\frac{\epsilon_h - h_{\min}(s)}{T_h} \right), \tag{2}$$

where $T_h > 0$ controls the transition width. The gate equals $1/2$ at $h_{\min}(s) = \epsilon_h$. It decreases as the margin grows and increases as the robot approaches or crosses a physical limit. Thus an estimated

violation does not reduce the request for guidance. The gate is smooth, so ϵ_h is its midpoint rather than a hard activation threshold.

Required interaction guidance. For the active event set, we use

$$w_e(s) = \max_{j \in \mathcal{J}(s)} \exp\left(-\frac{d_j(s)^2}{2\beta_j^2}\right), \tag{3}$$

where $\beta_j > 0$ sets the width in normalized event coordinates. Guidance increases as an active event target is approached and reaches one at that target. We define $w_e(s) = 0$ when no event is active. The active set matters here. Once an approach event is complete, its gate is disabled; any required contact maintenance is represented by its own condition. This prevents a completed grasp from leaving the approach gate active throughout the remaining motion.

Predicted failure guidance. The frozen predictor contributes

$$w_p^{(n)}(s) = \sigma\left(\frac{p_{\bar{\psi}_n}(s) - \eta}{T_p}\right), \tag{4}$$

where $T_p > 0$ controls the width and η is the predicted probability at which the gate equals $1/2$. Higher predicted risk raises this component even when the analytic margins are not small. Before enough valid rollout outcomes are available to fit the predictor, this component is disabled, and guidance uses the physical and event cues. The final demonstration weight is

$$w_n(s) = \max\left\{w_h(s), w_e(s), w_p^{(n)}(s)\right\}. \tag{5}$$

The maximum preserves a strong cue when the other two are small. For example, an approaching grasp can receive strong event guidance despite low predicted failure, while a small foothold margin can raise guidance without an active hand contact. Guidance is weak only when all three components are small. Their maximum is a supervision coefficient, not a calibrated probability of failure.

We use negative log likelihood for the supervised loss,

$$\mathcal{L}_{\mathrm{BGDL}}^{(n)}(\theta) = -\mathbb{E}_{(s^E, o^E, a^E) \sim \mathcal{D}_E}\left[\mathrm{sg}\big(w_n(s^E)\big) \log \pi_\theta(a^E \mid o^E)\right], \tag{6}$$

where sg stops gradients through the weight. Samples are drawn uniformly from the fixed demonstration set. We evaluate the current actor at the stored observation, score the recorded action, and multiply that sample's loss by its weight. The weight is not also used as a sampling probability. For a particular sample, the direct gradient contribution, apart from the averaging factor, is $-w_n(s_i^E)\nabla_\theta \log \pi_\theta(a_i^E \mid o_i^E)$. This expression identifies what BGDL changes in the update. The weight scales the supervised gradient, not the motor command. Consider two recorded states in a stair ascent. A sample near a tread edge can receive a larger coefficient than an approach sample with ample clearance and no other strong cue. Both remain in the dataset, and both are evaluated with the current actor. BGDL does not request a new expert action at a state encountered during a policy rollout.

3.4 Policy Optimization

The rollout path supplies task feedback from the actor's own behavior. The same physical cues and frozen predictor are evaluated at rollout states and their successors, but their outputs are used differently from the demonstration weights. For a transition entering s_{t+1}, the physical cost is

$$c_t = \sum_k v_k[-\bar{h}_k(s_{t+1})]_+ + v_F F_{t+1}, \tag{7}$$

where $[x]_+ = \max(x, 0)$, the coefficients are nonnegative, and F_{t+1} indicates terminal physical failure. Negative physical margins therefore incur a cost, with an additional penalty for a terminal failure. Desired contact proximity does not enter this cost. All transition quantities use the actual successor state before an automatic reset. We also use the bounded potential

$$\Phi_n(s) = -\kappa \max\{w_h(s), w_p^{(n)}(s)\}, \qquad \kappa \geq 0, \tag{8}$$

Algorithm 1 Boundary Guided Demonstration Learning

Require: Fixed demonstrations $\mathcal{D}_E$, initialized actor π_θ, critic V_ϕ, simulator, physical margins, and event rules

for training cycle $n = 0, 1, \ldots$ **do**
  Set $\theta_{\text{old}} \leftarrow \theta$ and hold the physical and event settings fixed.
  Freeze $p_{\bar{\psi}_n}$, or use $w_p^{(n)} = 0$ if a valid predictor is not yet available.
  Collect $\mathcal{D}_n$ with $\pi_{\theta_{\text{old}}}$, retaining final observations and termination flags.
  Evaluate rollout margins and risk to compute c_t, Φ_n, and $\widetilde{r}_t$.
  Compute rollout advantages and critic return targets.
  Evaluate the three cues at stored demonstration states to obtain fixed weights $w_n(s_i^E)$.
  Draw uniform demonstration minibatches separately from rollout minibatches.
  Update θ by maximizing Eq. equation 10.
  Update the critic parameters ϕ using the rollout return targets.
  Form valid future outcome labels and fit and calibrate the predictor for cycle $n + 1$.
end for
return Actor, observation preprocessing, and command mapping

which depends on physical margin and predicted failure. Required event proximity is excluded. Approaching a grasp or a seat should increase demonstration guidance without being treated as movement toward a physical violation. The reward used for the rollout update is

$$\widetilde{r}_t = r_t - \lambda_c c_t + \gamma(1 - z_t)\Phi_n(s_{t+1}) - \Phi_n(s_t), \tag{9}$$

where r_t is the task reward, $0 < \gamma < 1$, $\lambda_c \geq 0$, and $z_t = 1$ denotes genuine termination. The terminal mask removes the successor potential at a terminal transition. Administrative truncation instead retains the bootstrap value of the final observation and does not continue the return calculation into the reset episode. The potential is held fixed during rollout collection and the associated policy update. The actor maximizes

$$\mathcal{J}_n(\theta) = \mathcal{J}_{\text{PPO}}(\theta; \widetilde{r}) - \lambda_{\text{IL}}\mathcal{L}_{\text{BGDL}}^{(n)}(\theta) + \lambda_{\text{ent}}\mathbb{E}_{o\sim\mathcal{D}_n}[\mathcal{H}(\pi_\theta(\cdot \mid o))]\,. \tag{10}$$

Here $\mathcal{J}_{\text{PPO}}$ is the clipped actor objective formed from rollout actions and advantages computed using $\widetilde{r}_t$. The entropy term is written separately, and the imitation and entropy coefficients are nonnegative. Demonstration actions enter only $\mathcal{L}_{\text{BGDL}}^{(n)}$ and never enter the PPO importance ratio. Task learning therefore continues even when a demonstration sample receives a large weight.

The critic is trained separately from the actor. It uses rollout observations and privileged state information to estimate the return associated with $\widetilde{r}_t$. Its targets are computed from the rollout rewards and the bootstrap value where appropriate. Value estimates provide the baseline for the rollout advantages used in Eq. equation 10, while fitting the critic improves its predictions of the corresponding returns. The demonstration likelihood updates the actor, not the critic. Algorithm 1 gives these updates as separate steps.

After the actor and critic updates, completed rollout outcomes supply labels for the next predictor. A positive label requires an observed failure within the prediction window. An incomplete window without an observed failure is not automatically labeled safe. Fitting and calibration use separate trajectory splits, and the updated predictor is used in the next training cycle. The demonstrations remain fixed, although their risk-based weights can change as the predictor is updated. Appendix D gives the outcome handling and optimization details. At deployment, the actor receives the processed robot observation, and its command is mapped to the control targets. The critic, demonstration buffer, and guidance modules are not queried during execution.

3.5 Local Interpretation

The local analysis concerns when a policy action can inherit a physical condition satisfied by a reference action. It does not assume that the guidance weight reconstructs the admissible action set. Let h be a true physical margin, $\widehat{h}$ its estimate, and f a deterministic transition. Fix a state with $h(s) \geq 0$ and an action neighborhood containing a^E. Suppose the estimation error is at most ε_b at

the current and reachable successor states, and $\widehat{h}(f(s,\cdot))$ is locally Lipschitz with constant $L > 0$. For $0 < \alpha < 1$, assume the reference provides the reserve

$$\widehat{h}(f(s,a^E)) \geq (1-\alpha)\widehat{h}(s) + m, \qquad m > (2-\alpha)\varepsilon_b.$$

Under these assumptions, Theorem 2 in Appendix C gives

$$\|a - a^E\|_2 \leq \frac{m-(2-\alpha)\varepsilon_b}{L} \quad \Longrightarrow \quad h(f(s,a)) \geq (1-\alpha)h(s) \geq 0,$$

provided a lies in the same valid neighborhood. The sufficient deviation becomes smaller when the reference reserve decreases, local action sensitivity increases, or margin estimation becomes less accurate. This explains why closeness to a reference can matter in some physical states more than in others. BGDL uses its cues to assign imitation weights, but minimizing that loss does not enforce the radius at every state. The result is local, requires a suitable reference action, and does not establish safety over a full trajectory. The appendix gives the proof, stochastic extension, and limitations at changes in contact mode.

4 Experimental Setup and Evaluation

We evaluate BGDL across five whole-body humanoid tasks in simulation and on a physical Unitree G1. Our experiments are designed to measure task performance, learning efficiency, robustness under shifted conditions, and the effect of where demonstration supervision is applied.

Tasks and Demonstrations. We evaluate stair climbing, curb traversal, ground object pick up, chair sitting, and carrying, using 100 demonstrations per task. Each method uses five independent training seeds and 10 million transitions per policy, with evaluation every 200,000 transitions and 100 test episodes per policy and task. Training, validation, and test sets are separated by scene and trajectory. We evaluate standard and shifted conditions, including changes in geometry, payload, sensing, and dynamics. Task examples appear in Fig. 2; definitions and evaluation ranges are given in Appendix B.

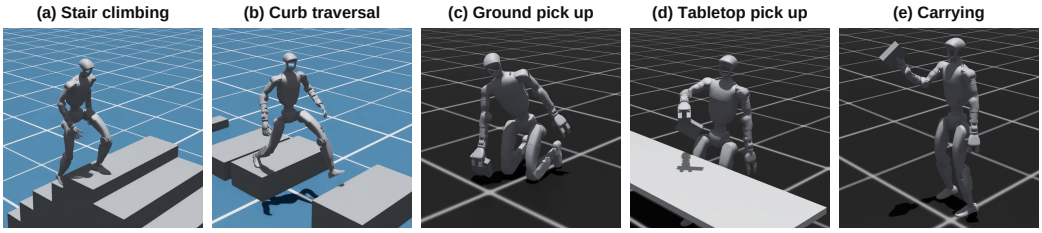

Figure 2: **Task Evaluation.** Terrain traversal, object interaction at different heights, and carrying require accurate foot placement, contact formation, and balance.

Baselines and Matched Controls. We compare BGDL with PPO, behavior cloning followed by PPO, an AMP style motion prior, reference tracking with task reward, and Q filtered PPO using a separate action value estimator (Nair et al., 2018). To isolate guidance placement, **Uniform matched** assigns the mean BGDL weight within each task, while **Random matched** permutes the same weights across its demonstration states. Both retain BGDL's demonstrations, initialization, architecture, training budget, and total imitation weight. All methods share observation and action spaces, policy capacity, interaction budgets, and test episodes. Appendix B.2 gives the baseline definitions and additional controls.

Metrics. Success requires task completion within the time limit without a protective intervention. We report success, falls, physical violations, and success under distribution shift. Physical violations are scored independently of BGDL's failure predictor. Learning efficiency is measured by the normalized area under the success curve and T_{80}, the first evaluation checkpoint reaching 80% success. Runs that never reach this threshold are censored at the training limit. Uncertainty is estimated by resampling training seeds and then episodes within each selected seed (Agarwal et al., 2021). Hyperparameters are selected on validation environments and fixed before testing. Further metric definitions appear in Appendix D.9.

5 RESULTS

5.1 OVERALL TASK PERFORMANCE

Table 1 summarizes performance across the five tasks. BGDL achieves **91.2%** average success, compared with 86.0% for reference tracking and 89.0% for Q-filtered PPO. Relative to reference tracking, this is a 5.2 percentage point improvement in success. The fall rate decreases from 12.8% to 8.0%, corresponding to a 37.5% relative reduction. BGDL also reaches the 80% success level earlier and retains higher success under the shifted evaluation conditions. The improvement is also consistent across individual tasks. BGDL reaches 94.0% success on stairs, 92.0% on curb traversal, 93.0% on object pick-up, 89.0% on sitting, and 88.0% on carrying. Relative to reference tracking, the largest improvement occurs on object pick-up, followed by stairs and curb traversal. The smaller gain on carrying suggests that the benefit of selective demonstration guidance varies with the demands of the task.

Method	Success (%) ↑	Falls (%) ↓	Shifted (%) ↑	T_{80} (M) ↓
PPO	67.2 ± 2.9	26.1 ± 2.0	48.6 ± 3.2	> 10
BC → PPO	75.8 ± 2.3	20.2 ± 1.9	57.4 ± 2.6	> 10
AMP-style prior	84.1 ± 1.8	14.1 ± 1.6	67.6 ± 2.1	6.8
Reference tracking	86.0 ± 1.7	12.8 ± 2.0	70.2 ± 2.2	6.0
Uniform matched	86.4 ± 1.9	11.4 ± 1.8	72.8 ± 2.3	5.6
Random matched	87.8 ± 1.8	10.6 ± 1.7	74.0 ± 2.2	5.6
Q-filtered PPO	89.0 ± 1.3	9.0 ± 1.2	76.0 ± 2.1	5.2
BGDL	$\mathbf{91.2 \pm 1.5}$	$\mathbf{8.0 \pm 1.1}$	$\mathbf{79.4 \pm 1.8}$	**4.4**

Table 1: **Average performance across five humanoid tasks.** Success, fall rate, and shifted success are reported as mean ± standard deviation across independent training runs. T_{80} denotes the first evaluation checkpoint at which the mean success curve reaches 80%, measured in millions of environment transitions. All methods use the same interaction budget.

5.2 LEARNING EFFICIENCY AND SUPERVISION PLACEMENT

Figure 3 compares learning under the same budget of 10 million environment transitions. The mean success curve for BGDL first reaches 80% at 4.4 million transitions, compared with 5.2 million for Q-filtered PPO and 6.0 million for reference tracking. The separation develops during training rather than appearing only at the final evaluation checkpoint. The matched controls test whether performance depends on where demonstration supervision is concentrated, rather than only on how much imitation is used. **Random matched** uses the same demonstration weights as BGDL but permutes them across demonstration states, while **Uniform matched** distributes the same total imitation weight uniformly along the demonstrations. Random matched reaches 87.8% success and Uniform matched 86.4%, compared with 91.2% for BGDL. Thus, preserving the overall amount of demonstration supervision is not sufficient to recover BGDL's performance; its placement along the trajectory also matters. In particular, concentrating supervision around states identified by restrictive physical conditions and required interactions performs better than distributing the same supervision uniformly or assigning it to unrelated states. Task-level learning curves are provided in Appendix B.3.

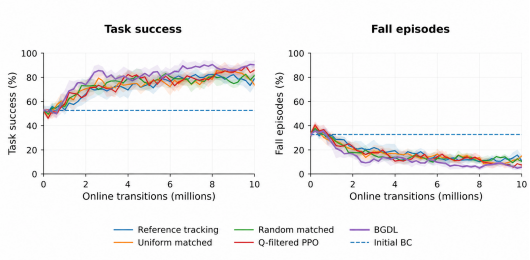

Figure 3: **Learning across five humanoid tasks.** Mean success and fall rate over five seeds; shading shows 95% confidence intervals.

5.3 WHICH SIGNALS MATTER?

Table 2 evaluates the contribution of the main BGDL training signals. Removing the failure-risk channel reduces success from 91.2% to 87.3% and shifted success from 79.4% to 73.0%. Because predicted failure is used both in demonstration weighting and in the shaping potential, this ablation measures the combined contribution of these two uses. Removing required-event guidance reduces

success to 88.2% and shifts success to 74.8%, suggesting that proximity to intended contacts provides information that is not captured by physical margins alone. The physical cost acts through the rollout objective rather than through the placement of demonstration supervision. Removing it reduces success to 89.0% and increases the fall rate from 8.0% to 10.4%. This separates the role of geometry-conditioned demonstration guidance from the direct penalty on physical constraint violations during policy learning.

5.4 Generalization under Distribution Shift

BGDL reaches 79.4% success when terrain geometry, object configuration, payload, and control conditions are changed beyond the training range. Reference tracking reaches 70.2%, Random matched reaches 74.0%, and Q filtered PPO reaches 76.0%. BGDL therefore retains 87.1% of its standard test success under the combined shift. The failure predictor also degrades under these changes, consistent with the change in the state distribution encountered by the policy. Its discrimination and calibration results are reported in Appendix B.5. This behavior is important for how the predictor should be interpreted: it is used to guide training, not as an execution time certificate of physical safety.

Variant	ID ↑	Falls ↓	Shifted ↑
No risk channel	87.3	11.7	73.0
No event guidance	88.2	10.2	74.8
No physical cost	89.0	10.4	76.5
BGDL	**91.2**	**8.0**	**79.4**

Table 2: **Ablation of BGDL training signals.**

5.5 Physical Robot Evaluation

We transfer the simulation-trained policies to a Unitree G1 without policy fine-tuning. The actor and observation preprocessing are kept fixed, and the policy receives only observations available on the physical robot. Control runs at 50 Hz. Protective interventions and aborted executions are counted as failures. Each method is evaluated in ten trials on each of the five tasks, giving 50 trials per method. BGDL completes **41 of 50 trials** successfully (**82.0%**), compared with 36 of 50 for reference tracking (**72.0%**). BGDL also requires fewer protective interventions across the evaluation, four compared with seven for reference tracking. Across tasks, BGDL succeeds in 9 of 10 stair trials, 9 of 10 curb trials, 8 of 10 object pick-up trials, 8 of 10 sitting trials, and 7 of 10 carrying trials. The 95% Wilson confidence interval for its aggregate success rate is 69.2–90.2%. These results show that the BGDL policy transfers from simulation to the Unitree G1 without policy fine-tuning across the evaluated tasks.

Task	Ref. success	BGDL success	BGDL interv.
Stairs	8/10	9/10	0
Curbs	7/10	9/10	0
Pick-up	7/10	8/10	1
Sitting	7/10	8/10	1
Carrying	7/10	7/10	2
Total	**36/50**	**41/50**	**4**

Table 3: **Physical evaluation on the Unitree G1.** Successes are reported over ten trials per task. Protective interventions and aborted executions are counted as failures.

6 Conclusion

We introduced Boundary Guided Demonstration Learning (BGDL), which allocates demonstration guidance according to local action freedom. BGDL uses physical margins, required interactions, and near term failure information to strengthen guidance near foothold limits, balance boundaries, and required contacts, while allowing adaptation where several actions are valid. Across five whole-body humanoid tasks, BGDL improves success, reduces falls, and reaches strong performance with fewer environment interactions than standard demonstration-based alternatives. Matched controls show that changing guidance placement lowers performance even when demonstrations and total imitation weight remain fixed. These results support allocating supervision according to local physical freedom. The learned policies also transfer from simulation to the Unitree G1 without policy fine-tuning. BGDL is intended as a way to use demonstrations more selectively during learning, rather than as a runtime safety mechanism. We discuss its scope and limitations in Appendix G.

AI USE STATEMENT

Generative AI tools were used to assist with writing and editing the paper, including improving clarity, grammar, organization, and presentation. All AI-assisted text was reviewed and revised by the authors, who verified the technical content, citations, and final wording and take full responsibility for the submitted paper.

REPRODUCIBILITY STATEMENT

We provide the details needed to reproduce the method and evaluation in the main paper and appendix. Section 3 describes the BGDL objectives and training procedure, while Appendix C contains the mathematical assumptions and complete proofs. The experimental setup is described in Appendix B, including the task definitions, data splits, and evaluation conditions. Appendix B.2 gives the baseline implementations and matched controls, and Appendix D specifies the observation and action interfaces, physical margins, required events, failure labels, predictor updates, optimization procedure, and terminal handling. Figure 5 summarizes the network architecture and information flow. Training settings and evaluation metrics are listed in Appendices D.8 and D.9. Additional results are provided in the appendix. Per-task learning curves are reported in Appendix B.3, and Appendix F includes the studies of action sensitivity, guidance placement, predictor transfer, robustness, and computational cost. Recorded evaluation sequences are shown in Figure 2. Section 4 specifies the training budgets, independent random seeds, evaluation counts, and the use of separate validation and test environments. The supplementary material also includes source code, experiment configurations, evaluation scripts, and videos of the robot experiments.

REFERENCES

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.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.

Haoyuan Cai, Zhenghao Peng, and Bolei Zhou. Robot-gated interactive imitation learning with adaptive intervention mechanism. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 6243–6256. PMLR, 2025.

Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive whole-body control for humanoid robots. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.107.

Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. HumanPlus: Humanoid shadowing and imitation from humans. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2828–2844. PMLR, 2024.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris M. Kitani, Changliu Liu, and Guanya Shi. OmniH2O: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 1516–1540. PMLR, 2024a.

Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning human-to-humanoid real-time whole-body teleoperation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8944–8951. IEEE, 2024b.

Tairan He, Jiawei Gao, Wenli Xiao, Yuanhang Zhang, Zi Wang, Jiashun Wang, Zhengyi Luo, Guanqi He, Nikhil Sobanbabu, Chaoyi Pan, Zeji Yi, Guannan Qu, Kris Kitani, Jessica Hodgins,

Linxi "Jim" Fan, Yuke Zhu, Changliu Liu, and Guanya Shi. ASAP: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. In *Proceedings of Robotics: Science and Systems*, 2025a.

Tairan He, Wenli Xiao, Toru Lin, Zhengyi Luo, Zhenjia Xu, Zhenyu Jiang, Jan Kautz, Changliu Liu, Guanya Shi, Xiaolong Wang, Linxi Fan, and Yuke Zhu. HOVER: Versatile neural whole-body controller for humanoid robots. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9989–9996, 2025b. doi: 10.1109/ICRA55743.2025.11128549.

Ryan Hoque, Ashwin Balakrishna, Ellen Novoseller, Albert Wilcox, Daniel S. Brown, and Ken Goldberg. ThriftyDAgger: Budget-aware novelty and risk gating for interactive imitation learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 598–608. PMLR, 2022.

Bingyi Kang, Zequn Jie, and Jiashi Feng. Policy optimization with demonstrations. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2469–2478. PMLR, 2018.

Jialong Li, Xuxin Cheng, Tianshu Huang, Shiqi Yang, Rizhao Qiu, and Xiaolong Wang. Amo: Adaptive motion optimization for hyper-dexterous humanoid whole-body control. *Robotics: Science and Systems 2025*, 2025a.

Jinhan Li, Yifeng Zhu, Yuqi Xie, Zhenyu Jiang, Mingyo Seo, Georgios Pavlakos, and Yuke Zhu. OKAMI: Teaching humanoid robots manipulation skills through single video imitation. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 299–317. PMLR, 06–09 Nov 2025b.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. MimicGen: A data generation system for scalable robot learning using human demonstrations. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1820–1864. PMLR, 2023.

Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6292–6299, 2018. doi: 10.1109/ICRA.2018.8463162.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. AWAC: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the 16th International Conference on Machine Learning*, pp. 278–287. Morgan Kaufmann, 1999.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. DeepMimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics*, 37(4), 2018. doi: 10.1145/3197517.3201311.

Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. AMP: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics*, 40 (4), 2021. doi: 10.1145/3450626.3459670.

Jerry Pratt, John Carff, Sergey Drakunov, and Ambarish Goswami. Capture point: A step toward humanoid push recovery. In *2006 6th IEEE-RAS international conference on humanoid robots*, pp. 200–207. Ieee, 2006.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Proceedings of Robotics: Science and Systems*, 2018. doi: 10.15607/RSS.2018.XIV.049.

Alexander Robey, Haimin Hu, Lars Lindemann, Hanwen Zhang, Dimos V. Dimarogonas, Stephen Tu, and Nikolai Matni. Learning control barrier functions from expert demonstrations. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3717–3724, 2020. doi: 10.1109/CDC42340.2020.9303785.

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

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

Jean Pierre Sleiman, Mayank Mittal, and Marco Hutter. Guided reinforcement learning for robust multi-contact loco-manipulation. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 531–546. PMLR, 2025.

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. doi: 10.1109/LRA.2021.3070252.

Tianyi Xie, Haotian Zhang, Jinhyung Park, Zi Wang, Bowen Wen, Jiefeng Li, Xueting Li, Qingwei Ben, Haoyang Weng, Yufei Ye, David Minor, Tingwu Wang, Chenfanfu Jiang, Sanja Fidler, Jan Kautz, Linxi Fan, Yuke Zhu, Zhengyi Luo, Umar Iqbal, and Ye Yuan. GRAIL: Generating humanoid loco-manipulation from 3D assets and video priors. *arXiv preprint arXiv:2606.05160*, 2026. doi: 10.48550/arXiv.2606.05160.

Lizhi Yang, Blake Werner, Massimiliano de Sa, and Aaron D. Ames. CBF-RL: Safety filtering reinforcement learning in training with control barrier functions. *arXiv preprint arXiv:2510.14959*, 2025.

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 *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 2143–2154. PMLR, 2025.

Chong Zhang, Wenli Xiao, Tairan He, and Guanya Shi. WoCoCo: Learning whole-body humanoid control with sequential contacts. In *Proceedings of the 8th Conference on Robot Learning*, 2024.

A Additional Related Work

Reinforcement learning from demonstrations. Demonstrations have been used both to initialize policies and to guide policy optimization. DAPG augments policy gradients with supervised expert updates, while POfD encourages the policy occupancy distribution to remain close to demonstrations (Rajeswaran et al., 2018; Kang et al., 2018). AWAC weights behavior likelihood updates using estimated advantage (Nair et al., 2020). BGDL studies a different question: how the influence of a fixed demonstration set should change across states as the physical freedom of the robot changes.

Demonstration generation and contact structure. MimicGen expands a small collection of manipulation demonstrations across new scene configurations (Mandlekar et al., 2023). GRAIL generates humanoid interaction trajectories from 3D scenes and video priors and retargets them to robot compatible motions (Xie et al., 2026). Contact structured methods such as WoCoCo use task contacts to simplify long horizon whole body learning (Zhang et al., 2024). These methods improve how demonstrations or contact structure are obtained. BGDL instead determines where stored demonstrations should have the strongest effect during policy learning.

Safety and recovery. Constrained Policy Optimization limits expected constraint cost during policy search (Achiam et al., 2017). Recovery RL learns when to invoke a recovery policy (Thananjeyan et al., 2021). Barrier based methods use explicit constraint models during training or execution (Robey et al., 2020; Yang et al., 2025). BGDL does not use its geometry signals as a runtime safety filter. They are used to allocate demonstration supervision during training. The shaping term follows the standard potential based result of Ng et al. (1999) and is separate from the local safety analysis below.

B Additional Experimental Details and Results

B.1 Task Definitions and Data Splits

We evaluate stair climbing, curb traversal, ground object pick up, chair sitting, and object carrying. Table 4 gives the task variations and completion criteria. Success requires completing the task within the episode horizon without a protective intervention.

Training, validation, and test data are separated by complete trajectories and environment instances. All frames from a trajectory remain in the same split, and closely related scene configurations are kept together. All methods use the same scene splits and test configurations. Methods that use demonstrations receive the same training trajectories. Demonstration construction and use are described in Appendix D. The standard test set contains unseen scene instances with configurations drawn from the training ranges. Shifted evaluation uses the expanded ranges in Table 4, varying terrain geometry, object pose, chair configuration, and payload. Training settings and evaluation metrics are given in Appendices D.8 and D.9, respectively.

Table 4: **Task ranges and success criteria.** Ranges are reported as training / shifted test.

Task	Geometry and interaction range	Success criterion
Stairs	rise $[0.10, 0.20]/[0.08, 0.22]$ m, tread $[0.25, 0.35]/[0.22, 0.40]$ m	upper platform, no fall
Curb	height $[0.05, 0.18]/[0.03, 0.22]$ m, yaw $\pm 20^\circ / \pm 30^\circ$	cross and regain support
Pick up	object offset $\pm 0.20 / \pm 0.30$ m, yaw $\pm 20^\circ / \pm 35^\circ$	grasp and lift
Sitting	chair height $[0.40, 0.50]/[0.35, 0.55]$ m, yaw $\pm 15^\circ / \pm 25^\circ$	stable seated contact
Carrying	payload $[0.5, 2.0]/[0.5, 2.5]$ kg, target offset $\pm 0.20 / \pm 0.30$ m	reach target without drop

B.2 Baselines and Matched Controls

PPO optimizes the task reward and physical cost without demonstration supervision. **BC to PPO** initializes the actor through behavior cloning and removes the imitation loss when reinforcement learning begins. **AMP style** uses an adversarial motion prior during policy learning, while **reference tracking** retains an explicit tracking objective alongside the task reward.

The matched controls test whether the assignment of demonstration weights matters when their total is held fixed. At each scoring cycle, **Uniform matched** replaces every demonstration weight within a task with the mean BGDL weight for that task. **Random matched** permutes the same weights across the task's demonstration states. Both preserve the sum of weights, while the random control also preserves their empirical distribution and breaks the original assignment of weights to states.

These controls use the same scored demonstration samples and scoring schedule as BGDL, together with the same actor initialization, architecture, and training budget. Matching the weight sums controls the total coefficient assigned to demonstration supervision, but does not require equal gradient magnitudes. Each weight multiplies the loss of a different sample, whose gradient depends on the current policy and recorded action.

Q filtered PPO adapts the demonstration selection rule of Nair et al. (2018). A separate action value estimator compares the demonstrated action with the policy action at the same demonstration state. The imitation term is applied when the estimator assigns greater value to the demonstrated action. This provides a selection rule based on estimated return rather than BGDL's physical, contact, and failure cues.

All methods use the same actor capacity, observation and action spaces, environment interaction budget, validation budget, and number of test episodes.

B.3 Per Task Learning Curves

Figure 4 reports learning separately for each task and for the five-task average. The task-level curves complement the aggregate results in the main paper and show whether the final improvement is driven by one behavior or persists across different forms of locomotion and interaction.

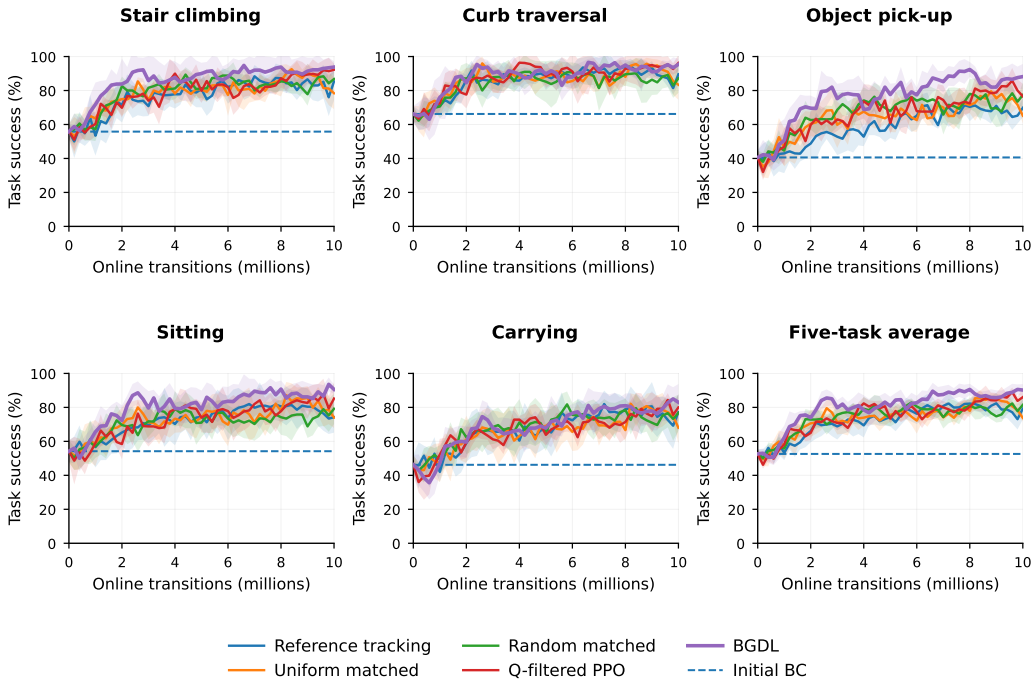

Figure 4: **Task-level learning.** Curves show the mean over five independently trained policies; shaded regions denote one standard deviation. The dashed line marks the behavior cloning initialization.

B.4 The Geometry Cues

The three BGDL signals capture different sources of restrictive action geometry. Table 5 evaluates each signal independently and in combination while keeping the learner, demonstration set, and training budget fixed. Physical margin guidance captures proximity to explicit constraints, required event guidance emphasizes precise task interactions, and failure prediction captures states whose difficulty is not fully described by either analytic signal.

Table 5: **BGDL guidance cues.** All variants use the same demonstration and environment interaction budgets.

Guidance cues	ID success ↑	Falls ↓	Shifted success ↑
Physical margin only	84.8	13.2	68.8
Required event only	83.9	14.5	67.5
Failure prediction only	85.7	11.9	71.0
Physical + event	87.3	11.7	73.0
Physical + failure	88.2	10.2	74.8
Event + failure	87.0	10.8	73.7
All three	**91.2**	**8.0**	**79.4**

B.5 Failure Predictor

We evaluate the failure predictor on complete trajectories that are excluded from both fitting and calibration. Table 6 reports discrimination and calibration in the standard test distribution and under the shifted evaluation conditions. The predictor remains informative under shift, although both ranking quality and calibration degrade as the policy encounters states farther from the training distribution.

Table 6: **Failure predictor evaluation.**

Split	AUROC ↑	AUPRC ↑	Brier ↓	ECE ↓
In distribution	0.91	0.74	0.098	0.011
Shifted	0.84	0.61	0.132	0.055

B.6 Recovery from Restrictive States

Task-level failure depends both on whether a policy enters a restrictive configuration and on how it responds once there. To evaluate the latter separately, we initialize each method from a shared set of physically admissible states near constraint boundaries. A trial is counted as a recovery when the policy returns above the fixed recovery margin within one second without falling or triggering another terminal physical violation. Table 7 reports recovery from 1,200 shared initial states. All methods are evaluated from exactly the same state. This evaluation isolates recovery after the policy has already reached a restrictive state, rather than success from the nominal task initialization. The CBF-based recovery method achieves the highest recovery rate at 83.2%, with BGDL close behind at 82.1%. This comparison is complementary to the end-to-end task results: BGDL is trained to improve behavior throughout the task by placing demonstration guidance selectively, whereas the recovery evaluation focuses only on behavior after initialization near a physical boundary.

Table 7: **Recovery from restrictive states.**

Method	Recovered	Recovery (%)
Reference tracking	869/1200	72.4
Uniform matched	904/1200	75.3
Random matched	941/1200	78.4
CBF-based recovery	998/1200	83.2
BGDL	985/1200	82.1

C Mathematical Details and Proofs

This section provides the formal results behind the analysis in the main paper. We develop the local margin argument for deviations from demonstrated actions and give supporting results for potential-based shaping, stochastic actions, weighted imitation error, and detached guidance weights. These results are local and conditional on the stated assumptions. They explain how physical margin and action sensitivity affect the allowable deviation around a demonstrated action, but they do not imply safety over an entire trajectory.

C.1 Boundary Representation and Safe Set

Let h_k denote a true physical margin for constraint k. The constraint set is

$$\mathcal{S}_{\text{safe}} = \bigcap_{k=1}^{K} \{s : h_k(s) \geq 0\}. \tag{11}$$

Membership means that the listed instantaneous constraints are satisfied. It does not mean that the robot can remain in this set under every future action. For continuous margins, $\partial\mathcal{S}_{\text{safe}} \subseteq \mathcal{S}_{\text{safe}} \cap \bigcup_k \{s : h_k(s) = 0\}$. Equality requires additional regularity. Even for one margin, a zero level set need not be a set boundary. For example, $h(s) = s^2$ on $\mathbb{R}$ has a zero at $s = 0$, although its nonnegative set is all of $\mathbb{R}$.

The training estimate $\widehat{h}_k$ need not equal h_k. We normalize it by a fixed positive scale c_k and keep required events separate,

$$\bar{h}_k(s) = \widehat{h}_k(s)/c_k, \qquad h_{\min}(s) = \min_k \bar{h}_k(s), \qquad d_j(s) \geq 0, \quad j \in \mathcal{J}(s). \tag{12}$$

Here d_j is the normalized distance to an active contact or affordance event, and $\mathcal{J}(s)$ is determined by task phase. An intended event has $d_j = 0$. Making that contact is not itself a violation. Forbidden collisions and violations during the contact are evaluated using separate physical criteria.

A failure predictor supplies $p_{\bar{\psi}_n}(s)$, the estimated probability of a specified failure within H steps under the recent collection policy. The index n identifies a frozen training target. The boundary representation is

$$\mathbf{b}_n(s) = \left(\{\bar{h}_k(s)\}_{k=1}^{K},\, \{d_j(s)\}_{j\in\mathcal{J}(s)},\, p_{\bar{\psi}_n}(s)\right). \tag{13}$$

It is not a scalar signed distance. In particular, rescaling $1/2 - p(s)$ does not turn a failure probability into geometric clearance. A prediction of $1/2$ has maximal Bernoulli entropy, but need not indicate uncertainty about the model parameters. The physical margins, event distances, and failure probabilities retain their own meanings.

C.2 Differentiable Aggregation of Physical Margins

The main method uses the hard minimum of normalized physical margins. The following soft minimum is an optional implementation variant, not an additional learned boundary or the default aggregator,

$$\widetilde{h}_{\tau_b}(s) = -\frac{1}{\tau_b} \log \sum_{k=1}^{K} \exp[-\tau_b \bar{h}_k(s)], \qquad \tau_b > 0. \tag{14}$$

Only physical margins enter this expression. It does not aggregate event distances with probabilities. Since the imitation weights are detached from the actor update, the main method does not require their aggregator to be differentiable.

Lemma 1 (Soft minimum approximation)**.** *For $m_h(s) = \min_k \bar{h}_k(s)$,*

$$m_h(s) - \frac{\log K}{\tau_b} \leq \widetilde{h}_{\tau_b}(s) \leq m_h(s). \tag{15}$$

Proof. Factor $\exp[-\tau_b m_h(s)]$ from the sum. The remaining factor is $S = \sum_k \exp[-\tau_b(\bar{h}_k(s) - m_h(s))]$. At least one term is one, and every term lies in $(0, 1]$, so $1 \leq S \leq K$. Substituting $\widetilde{h}_{\tau_b} = m_h - \log S/\tau_b$ proves both inequalities. □

The approximation is conservative relative to the estimated minimum, not necessarily relative to the true physical margin. It can be negative when all estimated margins are positive. Its offset depends on both K and τ_b. An ablation using it must disclose that offset and retain a fixed physical evaluator rather than interpreting its sign as a measured violation.

C.3 Boundary Guidance Weights

For thresholds ϵ_h, η and positive temperatures T_h, T_p, β_j, define

$$\begin{aligned} w_h(s) &= \mathrm{sigmoid}((\epsilon_h - h_{\min}(s))/T_h), \\ w_e(s) &= \max_{j \in \mathcal{J}(s)} \exp[-d_j(s)^2/(2\beta_j^2)], \\ w_p^{(n)}(s) &= \mathrm{sigmoid}((p_{\bar{\psi}_n}(s) - \eta)/T_p), \\ w_n(s) &= \max\{w_h(s), w_e(s), w_p^{(n)}(s)\}. \end{aligned} \tag{16}$$

The maximum over an empty event set is zero. A shrinking physical margin increases w_h, including after it becomes negative. A required event raises w_e without entering the safety cost. The maximum is a training weight, not a calibrated probability of failure.

For analysis, the critical subset is

$$B_{n,w_0} = \{s : w_n(s) \geq w_0\}, \qquad 0 < w_0 < 1. \tag{17}$$

This set can contain both intended contacts and unsafe states. Safety propositions apply only to its states satisfying their separate physical assumptions. A band $\{|h_{\min}(s)| \leq \epsilon\}$ may still be useful for plots of a particular physical margin, but it does not define all required contacts or all high risk states. A symmetric alternative weight $\exp(-|b_\psi(s)|/\beta)$ decreases again on the negative side as $|b_\psi|$ grows. That is unsuitable for interpreting its magnitude as increasing need for guidance during a physical violation. Neither this exponential nor the sigmoid gates implement a hard switch between imitation and reinforcement learning. Task reward remains active throughout training.

C.4 Policy Invariance under Fixed Potential Shaping

Let $c(s, a, s') \geq 0$ be the declared physical cost and write $\bar{r}(s, a, s') = r(s, a, s') - \lambda_c c(s, a, s')$, with fixed $\lambda_c \geq 0$. The potential excludes required-event proximity,

$$\Phi_n(s) = -\kappa \max\{w_h(s), w_p^{(n)}(s)\}, \qquad \kappa \geq 0. \tag{18}$$

It is bounded and held fixed during collection and policy optimization. At a true terminal state its value is zero. The corresponding reward is

$$\widetilde{r}_t = \bar{r}_t + \gamma(1 - z_t)\Phi_n(s_{t+1}) - \Phi_n(s_t), \tag{19}$$

where z_t indicates true termination. An administrative time limit is handled by a bootstrap, not by setting the underlying state's potential to zero.

Theorem 1 (Fixed-potential policy invariance)**.** *Assume $0 < \gamma < 1$, finite discounted returns, and a bounded fixed potential Φ that is zero at true terminal states. For $r^\Phi = \bar{r} + \gamma\Phi(s') - \Phi(s)$, every Markov policy satisfies*

$$V_\pi^\Phi(s) = V_\pi^{\bar{r}}(s) - \Phi(s), \qquad Q_\pi^\Phi(s, a) = Q_\pi^{\bar{r}}(s, a) - \Phi(s). \tag{20}$$

The optimal-policy sets for r^Φ and $\bar{r}$ are identical whenever an optimum exists. This is the potential-based shaping result of Ng et al. (1999), applied to the cost-penalized reward.

Proof. For a trajectory stopped after T transitions,

$$\sum_{t=0}^{T-1} \gamma^t r_t^\Phi = \sum_{t=0}^{T-1} \gamma^t \bar{r}_t + \sum_{t=0}^{T-1} \left[\gamma^{t+1}\Phi(s_{t+1}) - \gamma^t \Phi(s_t)\right] \tag{21}$$

$$= \sum_{t=0}^{T-1} \gamma^t \bar{r}_t - \Phi(s_0) + \gamma^T \Phi(s_T). \tag{22}$$

The last term is zero at a true terminal state. On a continuing trajectory it converges to zero as $T \to \infty$, since Φ is bounded. Conditioning on $s_0 = s$, or on $(s_0, a_0) = (s, a)$, gives Eq. equation 20. The subtraction is independent of the chosen action and of the continuation policy. Taking the supremum over policies therefore preserves the action ordering and the optimal-policy set. □

For a fixed initial-state distribution, the return difference is the same constant for every policy, including policies restricted to deployable observations. The theorem compares shaping with and without Φ while holding the remaining objective fixed. It does not establish that physical costs or imitation leave the unconstrained task optimum unchanged, or that PPO satisfies a constraint. It also does not prove convergence for an adaptive sequence of predictors.

The fixed potential value identities imply $A^{\Phi}_{\pi} = A^{\bar{r}}_{\pi}$ when values are exact. Thus shaping does not create a different true advantage. Any benefit to training must arise through approximation, initialization, or optimization. Resetting a truncated rollout without its continuation value leaves the residual $\gamma^T \Phi(s_T)$ in Eq. equation 22 and does not satisfy the same episodic argument.

C.5 Safety with Imperfect Boundary Estimation

In practice, the physical margin available to the learner is estimated and may differ from the true margin. We therefore analyze how bounded estimation error affects the amount by which a policy action may deviate from a demonstrated action while still preserving a one-step physical margin condition. Let h denote the true scalar physical margin, $\hat{h}$ its estimate, and f the deterministic transition model.

Theorem 2 (Local one step safety)**.** *Fix a state s with $h(s) \geq 0$, a reference action a^E, and a local action set $\mathcal{U}_s$ containing a^E. Assume*

$$|\widehat{h}(x) - h(x)| \leq \varepsilon_b \quad \text{for } x = s \text{ and every } x = f(s, a),\ a \in \mathcal{U}_s, \tag{23}$$

and, for $a_1, a_2 \in \mathcal{U}_s$,

$$|\widehat{h}(f(s, a_1)) - \widehat{h}(f(s, a_2))| \leq L\|a_1 - a_2\|_2, \qquad L > 0. \tag{24}$$

For $0 < \alpha < 1$, suppose the reference satisfies

$$\widehat{h}(f(s, a^E)) \geq (1 - \alpha)\widehat{h}(s) + m, \qquad m > (2 - \alpha)\varepsilon_b. \tag{25}$$

Every $a \in \mathcal{U}_s$ with

$$\|a - a^E\|_2 \leq r_s, \qquad r_s = \frac{m - (2 - \alpha)\varepsilon_b}{L} \tag{26}$$

satisfies

$$h(f(s, a)) \geq (1 - \alpha)h(s) \geq 0. \tag{27}$$

Proof. For any $a \in \mathcal{U}_s$,

$$\begin{aligned} h(f(s, a)) &\geq \widehat{h}(f(s, a)) - \varepsilon_b && (28)\\ &\geq \widehat{h}(f(s, a^E)) - L\|a - a^E\|_2 - \varepsilon_b && (29)\\ &\geq (1 - \alpha)\widehat{h}(s) + m - L\|a - a^E\|_2 - \varepsilon_b && (30)\\ &\geq (1 - \alpha)h(s) + m - L\|a - a^E\|_2 - (2 - \alpha)\varepsilon_b. && (31) \end{aligned}$$

The action condition makes the final reserve nonnegative. The final inequality in Eq. equation 27 uses $h(s) \geq 0$ and $0 < \alpha < 1$. □

With $\varepsilon_b = 0$, the result reduces to the exact-margin condition $\|a - a^E\|_2 \leq m/L$ in the supplied appendix. Setting $a = g_\theta(o)$ gives the deterministic-policy statement. For several constraints, the action must lie in the intersection of their valid neighborhoods and satisfy every radius bound. A positive margin for one constraint does not protect the others.

The factor $(2 - \alpha)\varepsilon_b$ accounts for error at both the current and successor states. If the current state is unsafe, the same algebraic lower bound may hold, but it does not imply recovery in one step. Contact mode changes can invalidate the local Lipschitz condition. A fitted predictor, residual quantile, or low training loss is not by itself a uniform error certificate. The theorem therefore does not turn the state weighted loss into a safety filter.

C.6 Probabilistic Safety under Stochastic Policies

Theorem 3 (Stochastic action bound)**.** *At a state satisfying Theorem 2, let $a \sim \pi_\theta(\cdot \mid o)$ be the action actually supplied to the controller. Define*

$$\zeta_s = \Pr(a \notin \mathcal{U}_s \mid s), \quad e_{1,s} = \mathbb{E}[\|a - a^E\|_2 \mid s], \quad e_{2,s} = \mathbb{E}[\|a - a^E\|_2^2 \mid s]. \tag{32}$$

When these moments are finite, the probability of violating the one step inequality is bounded by

$$\Pr\big(h(f(s,a)) < (1-\alpha)h(s) \mid s\big) \le \min\left\{1,\ \zeta_s + \min\left(\frac{e_{1,s}}{r_s}, \frac{e_{2,s}}{r_s^2}\right)\right\}. \tag{33}$$

Proof. Inside $\mathcal{U}_s$, Theorem 2 excludes a violation whenever $\|a - a^E\|_2 \le r_s$. The violation event is therefore contained in $\{a \notin \mathcal{U}_s\} \cup \{\|a - a^E\|_2 > r_s\}$. Apply the union bound and then Markov's inequality to the nonnegative random variables $\|a - a^E\|_2$ and $\|a - a^E\|_2^2$. The probability is at most one. □

When $\zeta_s = 0$ and $\varepsilon_b = 0$, the first moment bound becomes $\min\{1, Le_{1,s}/m\}$, which gives the exact-margin first moment result with a probability cap. The randomness here is in the action, not in the dynamics. Unmodeled disturbances need their own error or probability bound.

If an action distribution has mean $\mu_\theta(o)$ and finite covariance $\Sigma_\theta(o)$, then

$$e_{2,s} = \|\mu_\theta(o) - a^E\|_2^2 + \operatorname{tr}\Sigma_\theta(o). \tag{34}$$

Consequently, accuracy of the mean action does not control stochastic safety unless dispersion is also controlled. For clipped or squashed policies these moments and the local neighborhood are defined in executed-action coordinates, not obtained by silently substituting the unsquashed Gaussian parameters.

C.7 Boundary weighted Imitation and Conditional Error

Let μ_E be a distribution over reference tuples (s, o, a^E). It may be the uniform empirical distribution on the supplied demonstrations. For a fixed weight w_n, use $B = B_{n,w_0}$ from Eq. equation 17 and assume $p_B = \mu_E(B) > 0$. Define a nonnegative error ℓ_θ and

$$\mathcal{E}_w(\theta) = \mathbb{E}_{\mu_E}[w_n(s)\ell_\theta(s, o, a^E)]. \tag{35}$$

This is an error statistic, not a new name for the continuous action negative log likelihood objective.

Proposition 1 (Conditional error on critical states)**.** *If $\mathcal{E}_w \le \delta_{\mathrm{IL}}$, then*

$$\mathbb{E}_{\mu_E}[\ell_\theta \mid B] \le \frac{\mathcal{E}_w}{p_B w_0} \le \frac{\delta_{\mathrm{IL}}}{p_B w_0}. \tag{36}$$

Proof. Nonnegativity and $w_n \ge w_0$ on B give

$$\mathcal{E}_w \ge \mathbb{E}_{\mu_E}[\mathbf{1}_B w_n \ell_\theta] \ge w_0 \mathbb{E}_{\mu_E}[\mathbf{1}_B \ell_\theta] = p_B w_0 \mathbb{E}_{\mu_E}[\ell_\theta \mid B]. \tag{37}$$

Divide by $p_B w_0 > 0$. □

The factor p_B cannot be omitted. Consider 100 equally weighted samples, one in B, with weight one and squared error one on that sample and zero error on the remaining samples. The dataset-wide weighted error is 0.01, whereas the conditional boundary error is one. Omitting p_B would assert $1 \le 0.01$. The corrected bound is exact in this example.

If μ_E is empirical, the result concerns only that empirical distribution. It does not supply a reference at every policy visited state or a pointwise error bound. Reducing an upper bound on violation probability also does not prove that the actual violation probability decreases between training updates.

C.8 Likelihood Loss and Action Error

The implemented demonstration loss is

$$\mathcal{L}^{(n)}_{\mathrm{BGDL}} = -\mathbb{E}_{\mu_E}[w_n(s) \log \pi_\theta(a^E \mid o)]. \tag{38}$$

A probability density can exceed one, so its negative logarithm need not be nonnegative. Proposition 1 therefore does not apply directly to Eq. equation 38 for continuous actions.

For an unsquashed Gaussian with positive definite covariance $\Sigma(o)$ fixed with respect to the actor parameters during the update,

$$-\log \pi_\theta(a^E \mid o) = C(o) + \tfrac{1}{2}\Delta_\theta^\top \Sigma(o)^{-1}\Delta_\theta, \quad C(o) = \tfrac{1}{2}\log[(2\pi)^{d_a} \det \Sigma(o)], \tag{39}$$

where $\Delta_\theta = g_\theta(o) - a^E$ and here g_θ is the Gaussian mean. If $\lambda_{\max}(\Sigma(o)) \le \overline{\sigma}^2$ uniformly, then

$$\mathbb{E}_{\mu_E}[w_n \|\Delta_\theta\|_2^2] \le 2\overline{\sigma}^2 \left(\mathcal{L}^{(n)}_{\mathrm{BGDL}} - \mathbb{E}_{\mu_E}[w_n C(o)]\right). \tag{40}$$

To see this, diagonalize Σ. Each eigenvalue of Σ^{-1} is at least $1/\overline{\sigma}^2$, so $\|\Delta\|_2^2 \le \overline{\sigma}^2 \Delta^\top \Sigma^{-1}\Delta$. Substitute Eq. equation 39 and average.

The right-hand side uses the excess above the Gaussian normalization term, not raw negative log likelihood. If covariance is learned, that normalization term changes too; a decrease in raw likelihood loss alone does not establish a decrease in action error. Squashing or clipping requires an explicit change-of-variables or executed-action analysis. We therefore report measured squared action error separately from the training likelihood.

C.9 Additional Conditions for Policy visited States

A distributional comparison requires more than the empirical bound above. The following statement records the extra assumptions, rather than adding an expert-query mechanism to BGDL. For this subsection, μ_E is a reference *population* distribution, not a finite list of samples. Let C be a subset of B with $p_C = \mu_E(C) > 0$ on which the local safety assumptions hold. A reference a^E at the same state must be defined in both distributions.

Let ν_C be the normalized distribution of policy visited reference tuples conditioned on C. Assume absolute continuity and a bounded density ratio,

$$\frac{d\nu_C}{d\mu_E(\cdot \mid C)} \le K_C < \infty, \qquad r_s \ge r_* > 0 \quad \text{on } C. \tag{41}$$

Take $\ell_\theta = e_{2,s}$, the actual stochastic action error from Eq. equation 32, or its deterministic specialization. Then

$$\Pr_{\nu_C, \pi_\theta}(\text{one step inequality violation}) \le \min\left\{1,\ \mathbb{E}_{\nu_C}[\zeta_s] + \frac{K_C \mathcal{E}_w}{p_C w_0 r_*^2}\right\}. \tag{42}$$

Indeed, the density-ratio assumption gives $\mathbb{E}_{\nu_C}[\ell_\theta] \le K_C \mathbb{E}_{\mu_E}[\ell_\theta \mid C] \le K_C \mathcal{E}_w/(p_C w_0)$. Average the second moment bound in Theorem 3 and use $r_s \ge r_*$. This proves Eq. equation 42.

Outside C, the analysis provides no control. An unconditional bound must include at least the uncontrolled mass $\nu(C^c)$ in addition to the conditional bound weighted by $\nu(C)$. Empirical imitation error cannot replace $\mathcal{E}_w$ here without a separate statistical generalization argument. Continuous policy visitation is generally not absolutely continuous with respect to a finite empirical demonstration distribution. Thus Eq. equation 41 is not guaranteed by replaying the available data, and Eq. equation 42 is not a measured BGDL safety certificate.

C.10 Imitation Gradient Away from Critical States

Proposition 2 (Direct gradient attenuation)**.** *Fix a training cycle, stop gradients through w_n, and assume $\|\nabla_\theta \ell_{\mathrm{IL}}(\theta; s, o, a^E)\| \le G$ at a sample. If $w_n(s) \le \omega$ with $0 < \omega < 1$, then*

$$\|\nabla_\theta[\mathrm{sg}(w_n(s))\ell_{\mathrm{IL}}(\theta; s, o, a^E)]\| \le \omega G. \tag{43}$$

Proof. The weight is fixed in the actor update, so the gradient is $w_n(s)\nabla_\theta \ell_{\mathrm{IL}}$. Take its norm and apply both bounds. □

For the maximum gate, all active channels must be small. Sufficient conditions for $w_n(s) \le \omega$ are

$$\begin{aligned} h_{\min}(s) &\ge \epsilon_h + T_h \log \frac{1-\omega}{\omega}, \\ d_j(s) &\ge \beta_j \sqrt{2\log(1/\omega)} \quad \text{for every } j \in \mathcal{J}(s), \\ p_{\bar{\psi}_n}(s) &\le \eta - T_p \log \frac{1-\omega}{\omega}. \end{aligned} \tag{44}$$

They follow by inverting the sigmoid and exponential gates. For some parameters the last threshold is negative, so there is no state satisfying all conditions at that ω. The gates need not attain zero.

This is a bound on a sample's direct imitation gradient. It does not bound policy output changes elsewhere, accumulated optimization bias, or the final return. Shared network parameters and the reinforcement learning update can change actions even at states with negligible imitation weight.

D Implementation Details

D.1 Demonstration Construction and Use

For each task, we use 100 successful demonstration trajectories expressed in the same embodiment and control space as the learned policy. Each trajectory contains the robot state, the deployable observation, the executed action, and the successor state,

$$\tau^E = \left\{ (s_t^E, o_t^E, a_t^E, s_{t+1}^E) \right\}_{t=0}^{T_E-1}. \tag{45}$$

Here s_t^E contains the simulator information needed to compute training signals, o_t^E contains only quantities available to the actor, and a_t^E is represented in the same action coordinates used by π_θ. Reference poses are therefore not treated directly as action labels unless they have been mapped through the same controller interface.

We retain complete successful trajectories and remove runs that terminate because of a fall, forbidden collision, missed task contact, object drop, or manual reset. Demonstration, validation, and test scenes are separated before policy training so that trajectories from the same environment instance do not appear across splits.

Before reinforcement learning, the actor is initialized with behavior cloning,

$$\mathcal{L}_{\mathrm{BC}}(\theta) = -\mathbb{E}_{(o,a^E)\sim\mathcal{D}_E}\left[\log \pi_\theta(a^E \mid o)\right]. \tag{46}$$

The resulting checkpoint is used as the common initialization for BGDL and the demonstration-based comparison methods.

The demonstration set remains fixed during online training. At every actor update, we sample a demonstration minibatch uniformly from $\mathcal{D}_E$ and evaluate the current actor at the stored observation. For BGDL, the supervised contribution of each sample is then scaled by the detached guidance weight $w_n(s)$ according to Eq. equation 6. The weights affect the strength of the supervised update, not the demonstration sampling distribution. Demonstration actions are used only in the supervised objective. They are never inserted into the PPO importance ratio, and BGDL does not query an expert for actions at states visited by the current policy. Policy rollouts and demonstration replay therefore provide separate learning signals to the same actor.

D.2 Observation and Action Interface

The actor receives only observations that are available during deployment. The proprioceptive input contains base angular velocity, projected gravity, joint positions relative to the nominal configuration, joint velocities, the previous action, and the task command. Task-specific observations provide the local information required by the current behavior, such as relative terrain, object, chair, or target geometry. Privileged simulator state is used only during training. It provides the quantities needed

to compute physical margins, required interaction distances, failure labels, and critic targets. None of these privileged variables is added to the deployed actor observation.

The policy outputs normalized whole-body joint position offsets relative to the nominal. These values are mapped to joint targets around the nominal configuration and executed by the low-level joint controller. The same mapping is used when storing demonstration actions, so the demonstrated and policy actions share a common control representation. Action limits are applied after the policy output; demonstration losses and action error measurements are computed in the executed action coordinates. Observation normalization statistics are estimated from training data and frozen before evaluation and physical robot deployment.

D.3 Network Architecture

BGDL uses separate networks for policy learning, value estimation, and near term failure prediction. The actor and critic do not share parameters.

Actor. The actor is a multilayer perceptron with hidden dimensions $(512, 256, 128)$ and ELU activations. It receives the deployable observation o_t and parameterizes a diagonal Gaussian distribution over normalized joint targets,

$$\pi_\theta(a_t \mid o_t) = \mathcal{N}\left(\mu_\theta(o_t), \mathrm{diag}(\sigma_\theta^2)\right). \tag{47}$$

The mean is produced by the final linear layer, while the log standard deviation is learned during training. At evaluation, the policy mean is used for deterministic execution.

Critic. The critic uses a separate MLP with hidden dimensions $(512, 256, 128)$ and ELU activations. During simulation training it receives the policy observation together with privileged state information used for value estimation. This asymmetric actor critic design allows privileged information to improve learning without making it part of the deployed policy.

Failure predictor. The failure predictor is a separate MLP with hidden dimensions $(256, 128, 64)$ and SiLU activations. Its sigmoid output estimates the probability of a declared failure within the next H control steps,

$$p_\psi(s) = \sigma(f_\psi(s)). \tag{48}$$

The predictor shares no parameters with the actor or critic. It is updated from completed rollout outcomes and frozen while the actor is optimized in the following training cycle.

Figure 5 separates the deployment path from the components used during training. The critic uses privileged state information from policy rollouts to estimate values. Demonstration weights are computed at stored demonstration states using physical margins, required contact cues, and predicted failure. At deployment, the actor maps the processed robot observation o_t to the action a_t without querying the critic or guidance modules.

D.4 Physical margins and required events

Each physical cue has a fixed operational definition before training and evaluation. We normalize each estimated physical margin $\widehat{h}_k$ by a positive scale c_k,

$$\bar{h}_k(s) = \widehat{h}_k(s)/c_k, \qquad h_{\min}(s) = \min_k \bar{h}_k(s). \tag{49}$$

Foothold clearance. For a convex landing patch $\mathcal{P} = \{p : A_\ell^\top p \le b_\ell\}$ and a planned sole polygon with vertices v, we use

$$\widehat{h}_{\mathrm{foot}}(s) = \min_{\ell, v} \frac{b_\ell - A_\ell^\top (R_s v + p_s)}{\|A_\ell\|_2}. \tag{50}$$

This measures whole foot clearance rather than center point clearance. The active foot and landing patch follow the current contact phase.

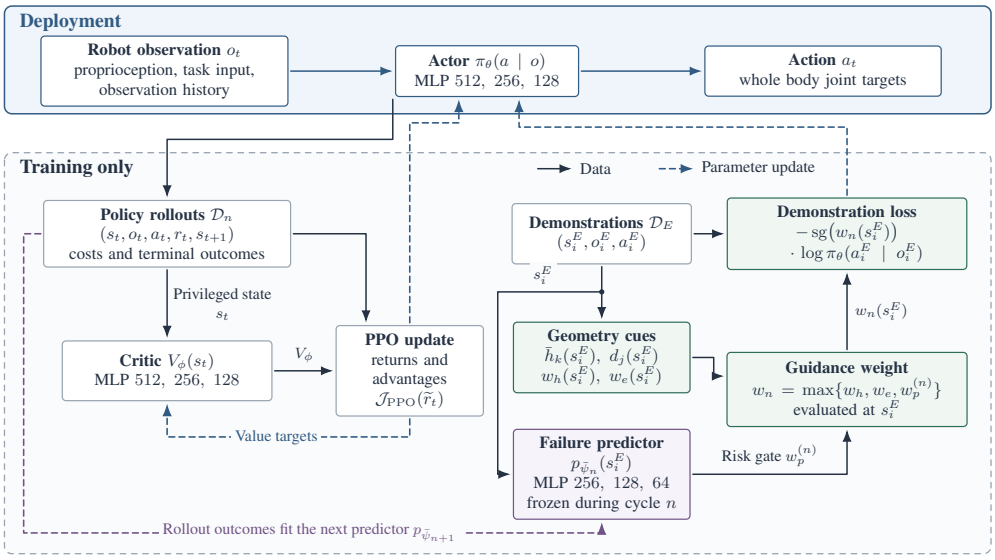

Figure 5: **BGDL network architecture and information flow.** Solid arrows show data or command flow; dashed arrows show parameter updates. PPO and weighted demonstration replay update the same actor. The demonstration loss is shown for one sample, with gradients stopped through its weight. Completed rollout outcomes train the failure predictor for the next cycle. The predictor remains fixed during each actor update.

Collision clearance. Let $\mathcal{F}(s)$ contain the body and environment pairs that are not allowed to contact in the current phase. We use

$$\widehat{h}_{\mathrm{coll}}(s) = \min_{(u,v)\in\mathcal{F}(s)} [\mathrm{sdist}(u,v) - d_{uv}^{\min}]. \tag{51}$$

Required hand object and body seat contacts are removed from $\mathcal{F}(s)$ only during the corresponding task phase.

Balance margin. We use a support based balance margin as a training cue. For the dynamic variant based on the planar capture point (Pratt et al., 2006),

$$\xi = c_{xy} + \dot{c}_{xy}/\omega, \qquad \omega = \sqrt{g/z_c}, \tag{52}$$

and the margin is the distance of ξ from the support polygon boundary. The reduced model is used only as a local cue. It is not treated as a general certificate for dynamic whole body motion.

Required event distance. For event j, let $\zeta_j(s)$ be the current interaction features, $\mathcal{Z}_j$ the valid target set, and W_j a positive definite scale. We define

$$d_j(s) = \min_{z\in\mathcal{Z}_j} \|W_j^{1/2}(\zeta_j(s) - z)\|_2. \tag{53}$$

The event is activated by the current task phase. Position, relative velocity, and local orientation error can be included when relevant.

D.5 Failure labels and predictor updates

Let $F_u = 1$ denote a failure on the transition entering s_u. The predictor target at s_t is

$$y_t^H = \mathbf{1}\{\exists u \in \{1,\dots,H\} : F_{t+u} = 1\}. \tag{54}$$

A state is positive as soon as a failure is observed inside its future window. A negative target requires a complete failure free window. Windows cut by a time limit, reset, or incomplete buffer segment are masked unless the outcome is already known.

Fitting and calibration are split by complete trajectories and scenes. The calibrated predictor is frozen before rollout collection and remains fixed during the actor update. Outcomes from that cycle are used only when training the predictor for a later cycle. Before enough valid outcome data exist, the failure guidance term is disabled and BGDL uses the physical and required event cues.

D.6 Training Procedure

The overall BGDL training loop is summarized in Algorithm 1 in the main paper. Here we provide the details that determine how each training cycle is carried out.

At the start of cycle n, the failure predictor used to compute $w_p^{(n)}$ is frozen. Policy rollouts are then collected with the current actor, and the resulting trajectories are used to compute physical costs, shaped rewards, advantages, and value targets. Demonstration minibatches are sampled uniformly from $\mathcal{D}_E$, and the state-dependent BGDL weight $w_n(s)$ scales each demonstration loss once. The actor is updated using the PPO objective together with the weighted demonstration objective, while the critic is optimized from rollout value targets. Completed rollout windows are added to the failure-prediction dataset after the policy update. Windows are labeled according to whether a declared failure occurs within the next H transitions; unresolved windows are excluded from the supervised predictor loss. A new predictor is then fitted and calibrated from the accumulated failure data and is used, while frozen, during the following training cycle.

D.7 Rollout Objective and Terminal Handling

For a transition entering s_{t+1}, the physical cost is

$$c_t = \sum_k v_k[-\bar{h}_k(s_{t+1})]_+ + v_F F_{t+1}. \tag{55}$$

Required contact proximity does not enter this cost. We use the true final state before an automatic reset.

With shaped reward $\widetilde{r}_t$, the temporal difference residual and generalized advantage estimate follow

$$\delta_t = \widetilde{r}_t + \gamma(1 - z_t)V_\phi(s_{t+1}) - V_\phi(s_t), \tag{56}$$

with standard GAE recursion (Schulman et al., 2015). A genuine terminal state stops value bootstrapping. A time limit uses the value of the final observation but does not continue the recursion into the reset episode. The clipped PPO objective follows Schulman et al. (2017).

D.8 Hyperparameters

We use the same PPO settings across all methods unless a baseline requires an additional objective. The actor and critic learning rates, discount factor, GAE parameter, PPO clipping range, entropy coefficient, and total interaction budget are fixed across tasks. BGDL specific parameters control three aspects of the method: λ_{IL} sets the overall strength of demonstration supervision, λ_c and κ determine the contribution of physical cost and potential shaping, and ϵ_h, T_h, η, T_p, and β_j determine how the three guidance signals are converted into state-dependent weights. Hyperparameters (Table 8) are selected on the validation environments and then kept fixed for test evaluation. Unless stated otherwise, the same BGDL settings are used for all five tasks.

The failure horizon of $H = 20$ steps corresponds to 0.4 s at the 50 Hz control rate. The margin and failure temperatures control how gradually the corresponding guidance channels increase around their thresholds; smaller values produce sharper transitions. We use a single default event width $\beta_j = 0.50$ for the reported experiments and keep the event definitions fixed within each task.

D.9 Evaluation metrics

Physical evaluation is kept independent of the learned failure predictor. A fixed evaluator records task completion, falls, forbidden collisions, object drops, and physical margin violations. This avoids scoring BGDL with the same model that determines its demonstration weights.

Table 8: **Training and BGDL guidance settings.**

Parameter	Value
Discount γ	0.99
GAE λ_{GAE}	0.95
PPO clip ϵ_{PPO}	0.20
Actor learning rate	3×10^{-4}
Critic learning rate	3×10^{-4}
Imitation coefficient λ_{IL}	0.50
Physical cost coefficient λ_c	1.00
Entropy coefficient λ_{ent}	10^{-3}
Potential scale κ	0.25
Failure horizon H	20 steps
Physical margin threshold ϵ_h	0.10
Physical margin temperature T_h	0.10
Failure temperature T_p	0.10
Failure threshold η	0.50
Default event width β_j	0.50
Control frequency	50 Hz
Training budget	10M transitions per task
Training seeds	5
Evaluation episodes	100 per seed and task

The boundary violation rate is

$$\mathrm{BVR} = \frac{\sum_i \sum_t V_{i,t}}{\sum_i T_i}, \tag{57}$$

where $V_{i,t}$ is a fixed physical violation indicator. Near-boundary recovery uses a shared state bank and fixed entry and recovery thresholds. For learning efficiency, T_{80} is the first evaluation checkpoint at which a training seed reaches 80% success. Runs that never reach the threshold are treated as censored at the final training budget.

E Additional Qualitative Results

E.1 Terrain Traversal

Figure 6 shows representative successful rollouts for stair climbing and curb traversal. In both cases, the robot must do more than reach the next contact. It must also complete the following support transfer while maintaining balance. The stair sequence shows progression from the lower steps to the upper surface, while the curb sequence shows the robot crossing the raised terrain and recovering a stable posture after landing.

Figure 7 shows failure cases from the reference tracking policy without BGDL. In the curb example, the robot loses lateral balance near the raised block and falls after the attempted crossing. In the stair example, a backward loss of balance occurs during the lower part of the ascent. Both failures occur around restrictive terrain contacts, where small deviations in foothold placement and support transfer can destabilize the motion.

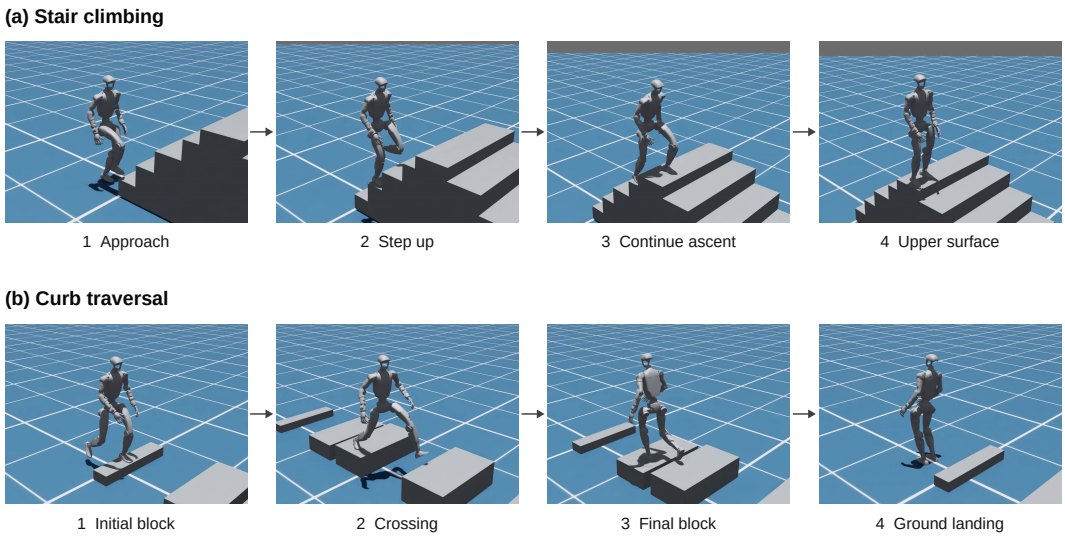

Figure 6: **Successful terrain traversal.** Stair climbing and curb traversal rollouts. Each row follows one episode from left to right and includes the support transfer after the critical terrain contact.

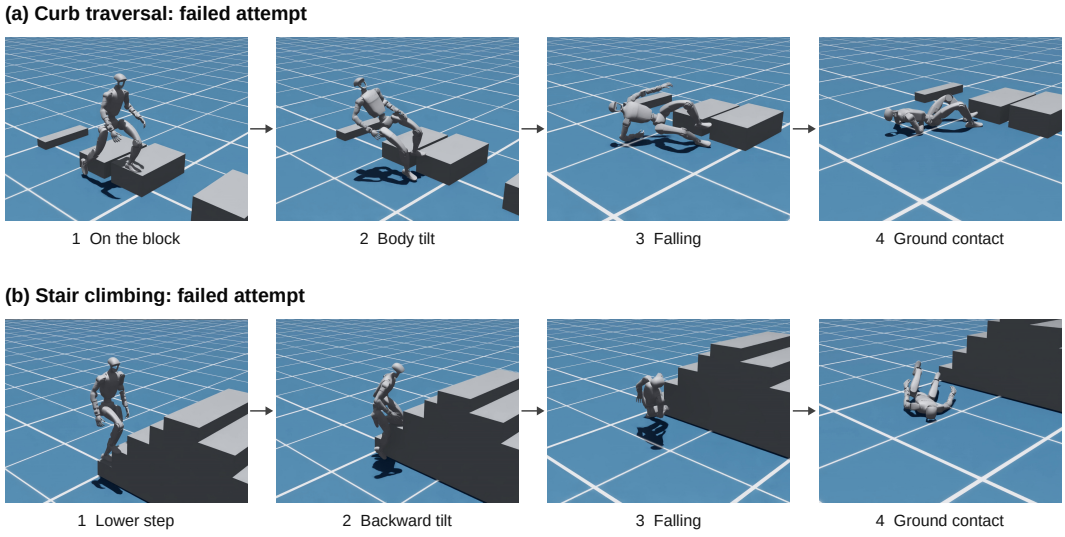

Figure 7: **Terrain traversal failures with reference tracking, without BGDL.** The curb attempt ends in a lateral fall after contact with the raised terrain, while the stair attempt shows a backward loss of balance during ascent. Both failures occur around restrictive terrain contacts, where accurate foothold placement and support transfer are critical.

E.2 Object Interaction

Figure 8 shows representative object interaction sequences at different heights. For ground pick-up, the robot reaches from a low posture, establishes contact, lifts the object, and returns toward an upright configuration while retaining the grasp. The tabletop example requires a different whole-body configuration, with the robot reaching forward, lifting the object from the surface, and stabilizing the grasp above the table. The carrying sequence starts after grasp establishment and shows the robot maintaining the object while stepping. Together, these examples highlight the different roles of precise interaction and post-contact adaptation: contact formation is locally restrictive, while a broader range of motions can become available once a stable grasp has been established.

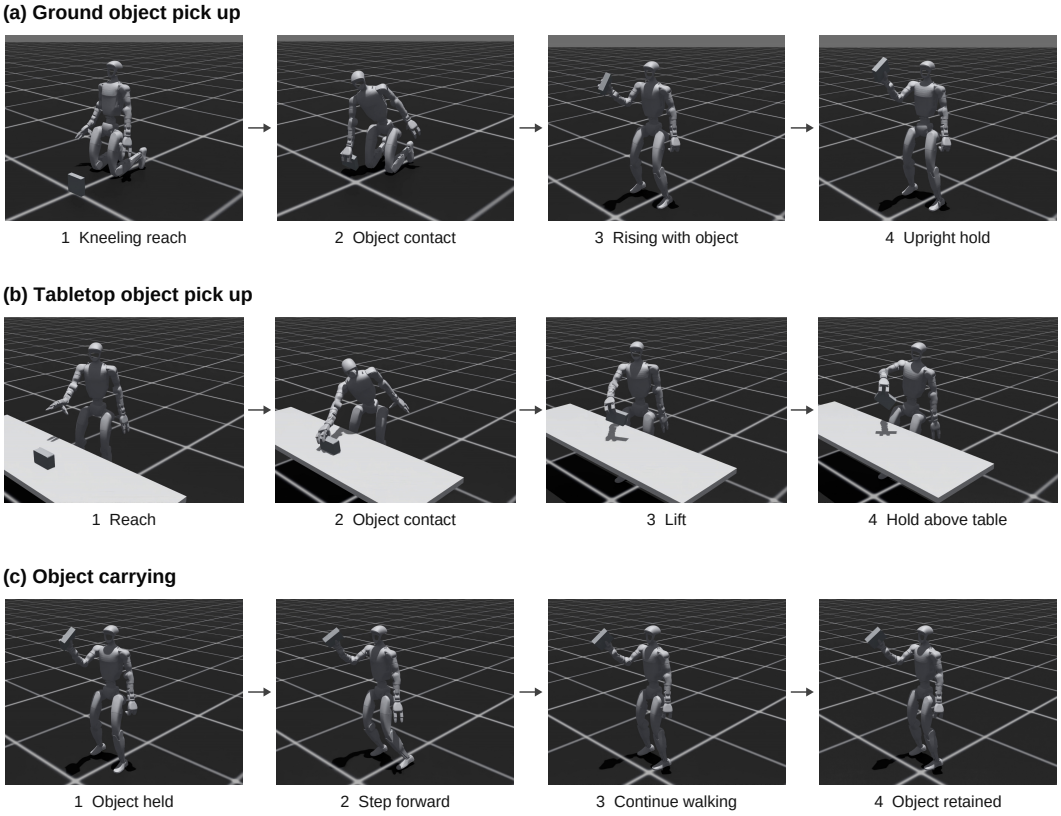

Figure 8: **Object interaction and carrying.** (a) Ground pick up from a low posture through lifting and recovery. (b) Tabletop pick up followed by an elevated hold. (c) Carrying after grasp establishment while maintaining object retention.

E.3 Additional Terrain Evaluation

Figure 9 provides an additional view of locomotion on the upper stair structure. The sequence begins after the robot has already reached the elevated portion of the terrain and therefore focuses on continued stepping rather than the complete ascent from the floor.

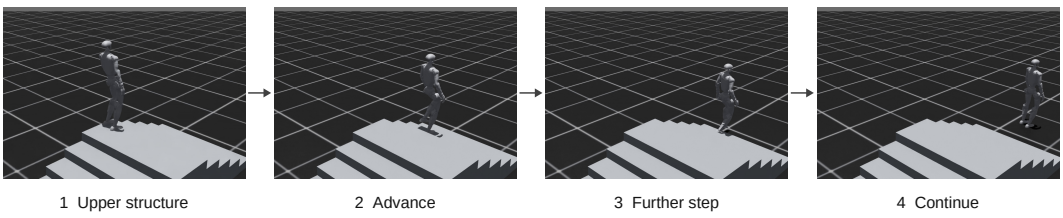

Figure 9: **Locomotion on the upper stair structure.** The sequence begins from an elevated configuration and shows continued stepping across the upper terrain.

F Additional Analysis of Demonstration Guidance

In this section we evaluate what the BGDL guidance weights capture and how much their placement matters during learning. We first measure local action tolerance by perturbing demonstrated commands from restored simulator states. We then test whether moving the same guidance weights

to different states changes policy learning, and finally evaluate predictor calibration, demonstration coverage, robustness, and training cost.

F.1 Local Action Tolerance and Guidance Weight

A small physical margin does not by itself imply that nearby action changes are consequential. We test this relation directly. A fixed set contains 60 valid simulation snapshots from each of two phases in each task, giving 600 states. Each snapshot includes the robot and object state, reference phase, observation history, and controller memory needed to reproduce the next command. States from the same trajectory remain grouped when estimating uncertainty. If a snapshot cannot be restored reliably, that failure is logged separately rather than being folded into the perturbation outcome.

At state s, perturb the reference action by

$$a^{(j)} = a^E + rDu_j, \qquad \|u_j\|_2 = 1, \tag{58}$$

where D specifies the units and relative scale of the action coordinates. The design uses 64 directions at each positive radius. The changed command is applied for one control interval, followed by a fixed controller for 0.4 seconds. The unperturbed continuation is a separate control, not 64 distinct probes. The snapshot is restored before each probe, and the direction bank and any continuation randomness are kept fixed across the compared cues. Figure 10 summarizes the action perturbation procedure.

We evaluate physical conditions and required interactions separately. For M probes, their joint acceptance fraction is

$$\widehat{q}_r(s) = \frac{1}{M}\sum_{j=1}^{M} \mathbf{1}\{\text{physical conditions and active task tests both pass}\}. \tag{59}$$

Required contacts are evaluated only when the corresponding event falls within the continuation window. Events scheduled for a later phase are therefore not counted as missed contacts. This finite continuation test is different from the one-step sufficient condition in Theorem 2.

For a latent actor, perturbation magnitude is measured after decoding, thresholding, and clipping in executed action coordinates. Distinct latent probes can therefore map to the same actuator target. The resulting statistic measures local action tolerance under the chosen perturbation distribution and continuation controller; it does not estimate the volume of the full feasible action set.

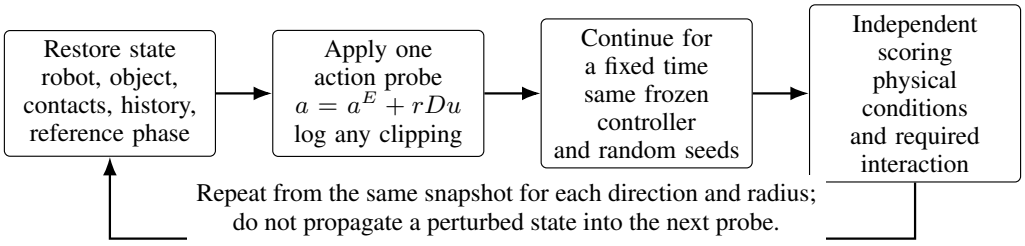

Figure 10: **Action perturbation study.** Each probe starts from the same saved state. A perturbed action is applied for one control step, after which the same controller is used for the continuation. Physical constraints and active task conditions are evaluated using fixed criteria. Before the next probe, the system is restored to the original state rather than continuing from the perturbed trajectory.

We compute this association within each task and phase so that it is not driven by coarse differences between behaviors. This analysis asks whether the weights track local action sensitivity; we test the effect of their placement on learning separately.

F.2 Where Demonstration Guidance Is Applied

The matched controls in the main experiments test whether the placement of demonstration guidance matters when the overall amount of supervision is held fixed. We evaluate this question more closely by asking whether BGDL distinguishes useful states even within the same task or contact phase.

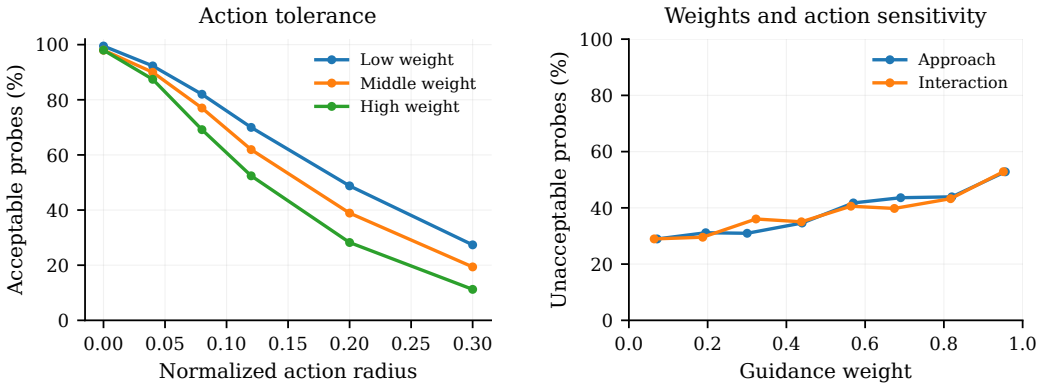

Figure 11: **Action tolerance and guidance weight.** Left, states are grouped by guidance weight within each task and phase before averaging. Right, rejection at radius 0.12 is shown separately for approach and interaction states.

Table 9: **Cue association with probe rejection.** Spearman correlations at radius 0.12 are computed separately in ten task and phase groups. The median and range summarize those groups, not independent trained models. Physical, event, and risk entries use the corresponding gates.

Cue	Median ρ	Minimum	Maximum
Physical margin	0.19	0.03	0.38
Required event	0.64	0.50	0.77
Failure score	0.10	-0.20	0.19
Full weight	0.69	0.48	0.77

Selective supervision based on risk has been studied previously (Hoque et al., 2022); our focus here is on whether the BGDL weights capture finer state-level differences associated with physical constraints and required interactions.

For task k and scoring cycle n, let $\mathcal{I}_k$ denote the corresponding demonstration indices. The task-level matched controls are

$$w_i^{\text{unif}} = \frac{1}{|\mathcal{I}_k|} \sum_{j \in \mathcal{I}_k} w_j, \qquad w_i^{\text{rand}} = w_{\sigma_k(i)}, \quad i \in \mathcal{I}_k, \tag{60}$$

where σ_k is a permutation within the task. The phase-shuffled control uses the same construction but restricts the permutation to states within the same task and contact phase. This preserves how much guidance each phase receives while removing its association with individual states.

All matched controls use the same demonstration samples and scoring schedule and preserve the total imitation weight within the corresponding matching group. The resulting gradient norms need not be identical, since the same weights are applied to different state-action pairs. Any data used to construct the scoring schedule is included in the common training budget.

We additionally evaluate a stricter within-trajectory control using 20 three-second trajectories for stair climbing and object pick-up. In this case, means and permutations are computed separately within each trajectory, and phase permutations remain within that trajectory. This control is reported separately from the task-level matching defined in Eq. 60.

Table 10 compares BGDL with the matched controls and alternative selection rules. In particular, phase-shuffled guidance provides a stricter comparison than task-level shuffling because it preserves the amount of supervision assigned to each contact phase. Any remaining difference from BGDL therefore reflects state-level placement within the phase, rather than simply identifying that a contact event is approaching.

We also test how sensitive learning is to the temporal placement of the guidance weights. We shift each weight sequence by -0.4 to 0.4 seconds along its demonstration trajectory, where a positive

Table 10: **Selector comparison.** Results are aggregated across five tasks and five training runs per task, with 100 evaluation episodes per run. Brackets denote pointwise 95% bootstrap intervals obtained by resampling runs within each task. Contact only and Q-filtered PPO are separate selection rules rather than permutations of the BGDL weight vector.

Weighting rule	Success (%)	Falls (%)
Reference tracking	86.0 [84.8, 87.4]	12.8
Uniform matched	86.4 [85.0, 87.6]	11.4
Random matched	87.8 [86.9, 88.8]	10.6
Phase shuffled	89.8 [88.8, 90.8]	8.8
Contact only	88.2 [87.1, 89.3]	9.8
Q filtered PPO	89.0 [88.0, 90.0]	9.0
BGDL	91.2 [89.6, 92.8]	8.0

offset assigns a weight to a later state. Circular indexing preserves the weight vector while changing its alignment with the demonstrated states. This experiment is intended as a controlled misalignment test rather than a model of observation delay. Demonstration actions, task rewards, physical costs, shaping terms, and loss coefficients remain unchanged. Figure 12 shows how success and fall rate change as the guidance weights are moved away from their original alignment.

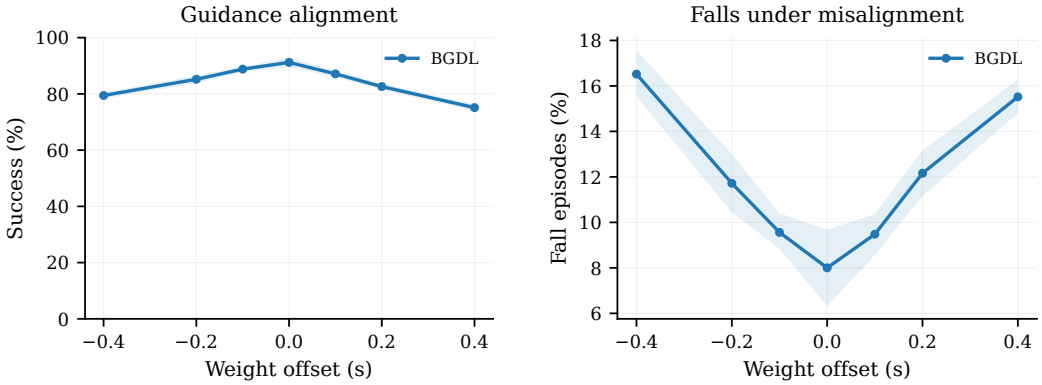

Figure 12: **Weight alignment.** Offsets change which demonstration states receive each weight without changing the weight distribution. Curves average tasks equally; bands denote pointwise 95% bootstrap intervals across training runs. Zero offset corresponds to the original BGDL alignment.

F.3 Guidance Around Contact Transitions

We study how the BGDL signals change around required interactions by aligning stored demonstrations to foot contact or grasp establishment. These traces are computed only from stored demonstrations and do not change the actions executed by the robot. At each observation, the current actor is evaluated against the recorded demonstration action. Once an approach event is completed, its event gate is disabled; contact maintenance, when needed, is handled by a separate condition. Figure 13 shows the physical, event, and failure guidance around the aligned contact transitions.

We also compare the size of the demonstration update with the PPO update. Both gradients are measured before gradient clipping and before the optimizer step. Let g_{IL} denote the weighted demonstration gradient before multiplication by λ_{IL}, and let g_{PPO} denote the rollout gradient. We define

$$R_g = \frac{\lambda_{\mathrm{IL}} \|g_{\mathrm{IL}}\|_2}{\|g_{\mathrm{PPO}}\|_2 + \varepsilon}, \qquad \varepsilon > 0. \tag{61}$$

The parameter subset, norm, and stabilizing constant are fixed across methods. We report the total guidance weight alongside this ratio because equal weight sums do not necessarily produce gradients of equal magnitude or direction. Figure 14 compares reference-action error with the relative size of the demonstration and PPO gradients.

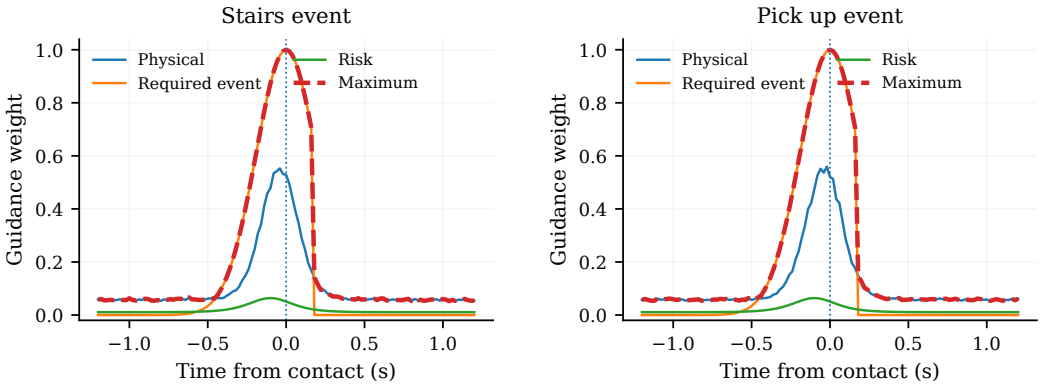

Figure 13: **Guidance around a required contact.** Weights are evaluated on stored demonstration states and aligned to the reference contact at $t = 0$. The event gate becomes inactive after the approach phase has completed.

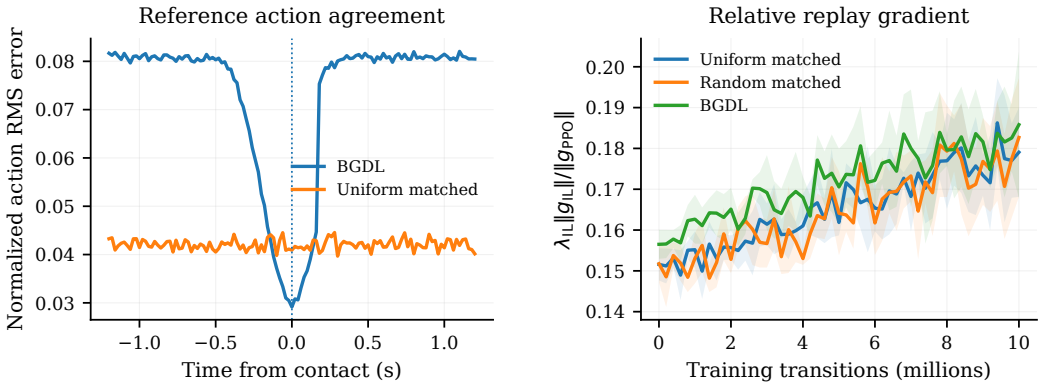

Figure 14: **Reference agreement and gradient magnitudes.** Left, normalized action error along a stored stair demonstration. Right, the norm of the scaled imitation gradient relative to the PPO gradient. Bands show one sample standard deviation across training runs.

A larger departure from the demonstrated action is useful only when task completion and motion quality remain acceptable. The recorded sequences in Appendix E.2 show contact, lifting, and object retention, but they are not paired executions from identical initial conditions. We therefore use them as qualitative examples rather than as evidence for differences between weighting rules or as the source of the action errors in Figure 14.

F.4 Completion, Falls, and Execution Time

We vary the imitation coefficient while keeping the actor architecture, demonstration set, physical costs, observations, and training budget fixed. Each coefficient is trained independently using the same validation allowance. The setting $\lambda_{\mathrm{IL}} = 0.5$ corresponds to the configuration used in the main experiments. At $\lambda_{\mathrm{IL}} = 0$, demonstration supervision is removed entirely.

A low fall rate is not necessarily useful if the robot simply avoids making progress. We therefore report both task success and the fraction of all episodes completed successfully by elapsed time t,

$$C(t) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{\text{episode } i \text{ completes successfully by } t\}. \tag{62}$$

Failed episodes remain in the denominator. We compute $C(t)$ separately for each task and training run and then average across tasks. Every attempt is assigned an outcome within the common 15-second horizon. Completion time conditioned on success is reported separately because excluding failed episodes measures a different quantity. Figure 15 shows the relationship between imitation strength, task success, physical failure, and completion time. This comparison helps separate improvements in task execution from policies that reduce falls mainly by behaving more conservatively. Operating points are selected on validation scenes before evaluation on the held-out test set.

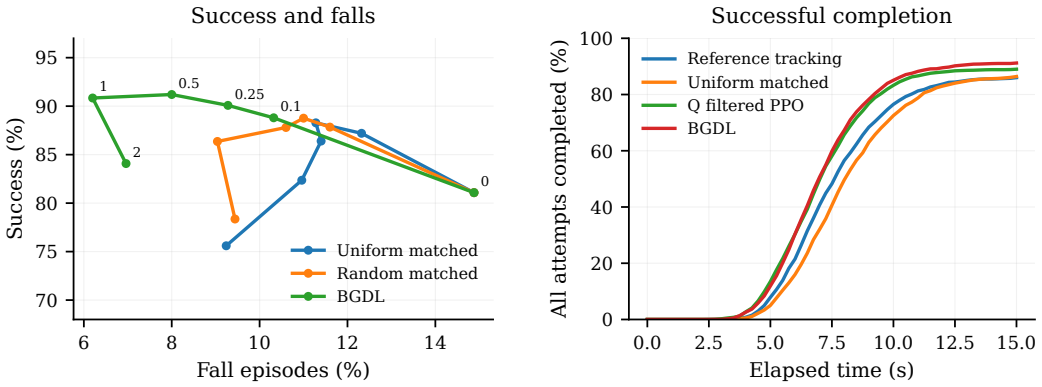

Figure 15: **Completion and physical failure.** Left, imitation strength varies from 0 to 2. Labels mark BGDL coefficients; connected points describe the sweep rather than a fitted Pareto frontier. Right, completion by elapsed time uses all evaluation episodes, including failed attempts in the denominator.

F.5 Failure Prediction Across State Sources

The failure predictor is fitted on policy rollouts but is also evaluated on stored demonstration states when computing BGDL weights. A successful expert continuation does not, by itself, show whether the current policy would fail from the same state. We therefore restore each state and continue from it using the policy being evaluated.

The study uses 120 states from each source and 20 continuations per state. States originating from the same trajectory remain in the same fitting, calibration, or evaluation split. Predictions are recorded before each continuation using the same future horizon as during predictor training. A failure within that horizon provides a positive outcome, while unresolved windows are censored. Only complete windows are used for the calibration results reported here. Calibration fitting and evaluation use separate trajectories (Guo et al., 2017).

With prediction p_i, n_i continuations, and k_i observed failures at state i, the Brier score over individual branch outcomes is

$$\text{Brier} = \frac{\sum_i [k_i(1-p_i)^2 + (n_i - k_i)p_i^2]}{\sum_i n_i}. \tag{63}$$

This differs from computing squared error against the empirical failure fraction k_i/n_i. Uncertainty estimates preserve grouping by source trajectory rather than treating individual continuations as independent training runs. Table 11 reports predictor performance across the different state sources.

Figure 16 shows the corresponding reliability curves for rollout and restored demonstration states. Figure 17 compares predictors and continuation policies from different stages of training.

F.6 Terrain Geometry, Payload, and Delay

We also test how the learned policies respond to changes in terrain geometry and execution conditions. For stair traversal, we vary tread depth and step height. For carrying, we vary payload and added action delay. Task definitions, permitted contacts, and evaluation criteria remain fixed, while task commands and references are updated consistently with the changed scene. The stair grid is

Table 11: **Prediction by state.** Each state is evaluated with 20 policy continuations. Failure is the observed failure fraction across those continuations. Brier score, log loss, and eight-bin ECE are computed from predictions and outcome counts.

State source	States	Failure	Brier	Log loss	ECE
Rollout nominal	120	0.148	0.097	0.323	0.006
Rollout shifted	120	0.198	0.141	0.446	0.049
Demonstration nominal	120	0.195	0.125	0.400	0.037

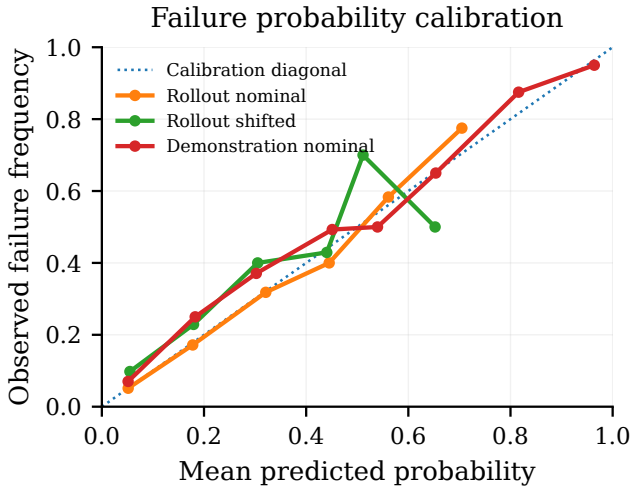

Figure 16: **Predictor reliability.** Points show mean prediction and observed failure fraction in nonempty bins. The final-stage predictor is evaluated on rollout states and restored demonstration states. Counts are retained with the data; the curve is not inferred from aggregate calibration scores.

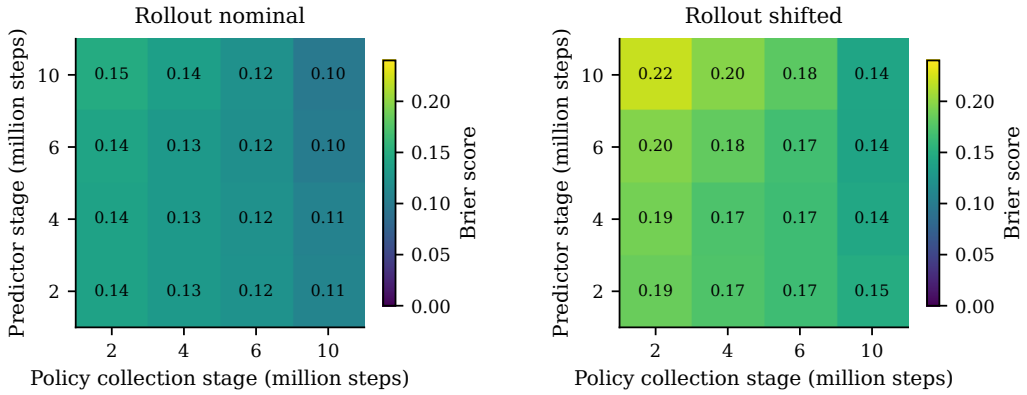

Figure 17: **Predictor transfer across training stages.** Rows fix the predictor stage and columns fix the policy supplying continuation outcomes. Both panels share the Brier scale. Evaluation of a later predictor on an earlier policy is retrospective and is not information available to that earlier policy update.

centered at a tread depth of 0.28 m and a step height of 0.12 m. Figure 18 compares performance across this geometry grid. Cells are interpreted using the training ranges given in Appendix B.

The carrying evaluation uses the nominal payload and zero added delay as its reference condition. Figure 19 shows performance as payload and action delay are varied.

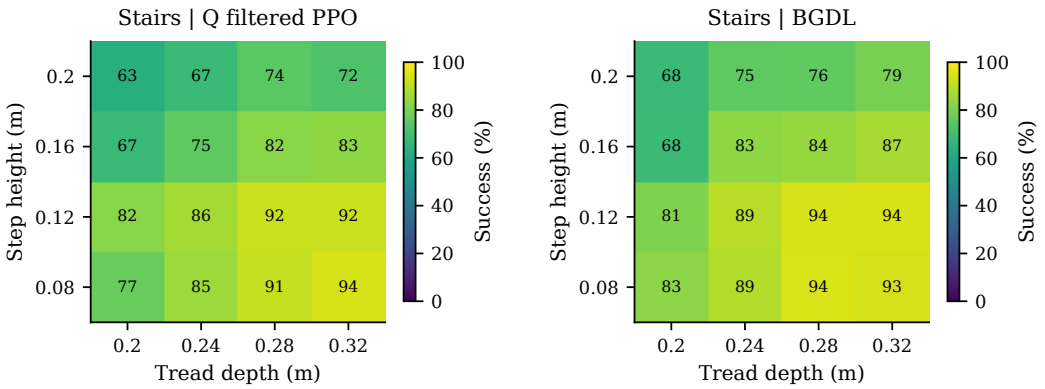

Figure 18: **Stair geometry comparison.** Each cell averages five independently trained policies with 100 evaluation episodes per policy. The panels share a 0–100% scale, without spatial interpolation or normalization by each method's nominal success.

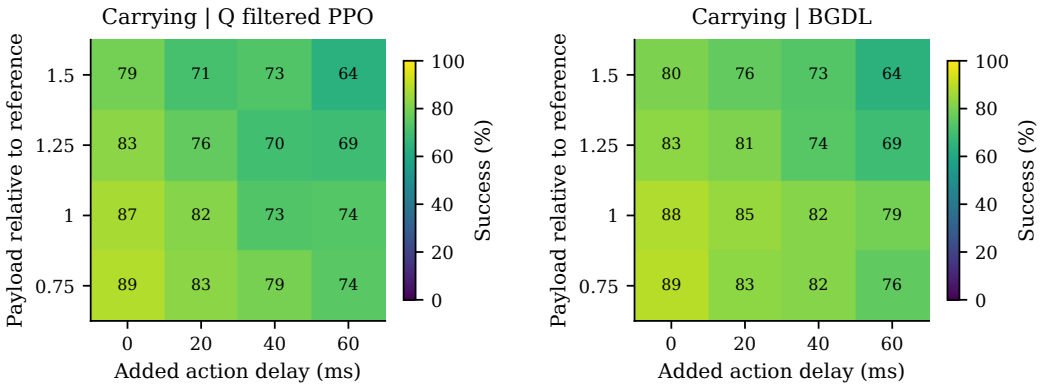

Figure 19: **Payload and added action delay.** Each cell uses the same number of trained policies and evaluation episodes as the stair grid. The nominal payload and zero added delay define the reference condition.

Training-time guidance errors and execution-time observation errors affect different parts of BGDL. Errors in the geometry cues change where demonstration supervision is applied during training, whereas execution noise acts directly on the frozen policy. Because the BGDL guidance signals are not evaluated at deployment, runtime robustness is determined by the actor's observation and control channels.

Figure 7 provides qualitative examples of several failure modes. The quantitative categories in Table 14 come from the recorded outcome logs rather than from visual interpretation of the image sequences.

F.7 Run-Level Results and Failure Outcomes

Aggregate scores weight each task equally. For independently trained task policies, uncertainty is estimated by resampling runs within each task. We do not treat identical seed indices across methods as paired experiments. For a multitask policy, the complete task vector from each run is kept together during resampling. Reporting run-level variation is especially important when only a small number of independent training runs are available (Agarwal et al., 2021).Figure 20 shows the task-level difference between BGDL and each baseline, with uncertainty estimated from the independent training runs. Table 12 reports the endpoint result for every training run. The corresponding learning

Table 12: **Individual endpoint results.** Entries report successes out of 100 evaluation episodes for each independently trained policy. Means and sample standard deviations are computed across the five runs.

Method	Task	1	2	3	4	5	Mean	SD
Reference tracking	Stairs	86	86	87	89	92	88.0	2.5
Reference tracking	Curbs	84	83	83	89	91	86.0	3.7
Reference tracking	Pick up	86	81	83	85	95	86.0	5.4
Reference tracking	Sitting	79	86	84	85	86	84.0	2.9
Reference tracking	Carrying	85	81	86	89	89	86.0	3.3
Uniform matched	Stairs	90	90	86	89	90	89.0	1.7
Uniform matched	Curbs	89	90	89	89	78	87.0	5.0
Uniform matched	Pick up	88	85	87	84	81	85.0	2.7
Uniform matched	Sitting	89	91	79	84	82	85.0	4.9
Uniform matched	Carrying	86	84	89	89	82	86.0	3.1
Random matched	Stairs	93	90	89	88	90	90.0	1.9
Random matched	Curbs	86	90	87	86	96	89.0	4.2
Random matched	Pick up	91	87	86	88	88	88.0	1.9
Random matched	Sitting	81	86	87	85	91	86.0	3.6
Random matched	Carrying	85	87	87	85	86	86.0	1.0
Phase shuffled	Stairs	89	91	91	92	97	92.0	3.0
Phase shuffled	Curbs	92	89	90	88	91	90.0	1.6
Phase shuffled	Pick up	93	90	91	87	94	91.0	2.7
Phase shuffled	Sitting	86	86	87	82	94	87.0	4.4
Phase shuffled	Carrying	89	88	89	87	92	89.0	1.9
Contact only	Stairs	89	92	90	92	92	91.0	1.4
Contact only	Curbs	91	86	87	85	91	88.0	2.8
Contact only	Pick up	90	85	87	88	90	88.0	2.1
Contact only	Sitting	87	94	90	89	80	88.0	5.1
Contact only	Carrying	83	85	86	86	90	86.0	2.5
Q filtered PPO	Stairs	94	94	91	92	89	92.0	2.1
Q filtered PPO	Curbs	90	91	88	91	95	91.0	2.5
Q filtered PPO	Pick up	88	93	86	93	90	90.0	3.1
Q filtered PPO	Sitting	84	86	85	83	87	85.0	1.6
Q filtered PPO	Carrying	94	86	87	85	83	87.0	4.2
BGDL	Stairs	93	99	95	98	85	94.0	5.6
BGDL	Curbs	98	94	89	93	86	92.0	4.6
BGDL	Pick up	89	92	93	93	98	93.0	3.2
BGDL	Sitting	86	89	90	81	99	89.0	6.6
BGDL	Carrying	87	92	86	90	85	88.0	2.9

Table 13: **Learning summaries.** Area uses success fractions over the common interaction horizon. Reached counts include five tasks and five training runs per task. A run that does not reach 80% success is censored at the end of training rather than assigned an extrapolated crossing time.

Method	Mean area	Reached 80%
Reference tracking	0.748	25/25
Uniform matched	0.750	25/25
Random matched	0.764	25/25
Phase shuffled	0.783	25/25
Contact only	0.769	25/25
Q filtered PPO	0.778	25/25
BGDL	0.799	25/25

curves are provided in Appendix B.3, while Table 13 summarizes learning efficiency over the same runs. Each unsuccessful episode is assigned one primary outcome category. When multiple physical events occur during the same episode, a fixed precedence rule determines the recorded outcome, so the categories remain mutually exclusive. Table 14 gives the breakdown of terminal outcomes across the evaluated methods.

Table 14: **Failure outcomes.** Counts cover 2,500 evaluation episodes per method. Support, toe, and collision entries denote terminal falls; contact, object, and timeout entries denote other failure outcomes. Object loss occurs only in tasks involving an object. The categories are taken from the recorded outcome logs rather than inferred from the frame sequences.

Method	Support	Toe	Collision	Contact	Object	Timeout	Total
Reference tracking	153	45	122	11	3	16	350
Q filtered PPO	106	15	104	24	8	18	275
BGDL	92	16	92	10	3	7	220

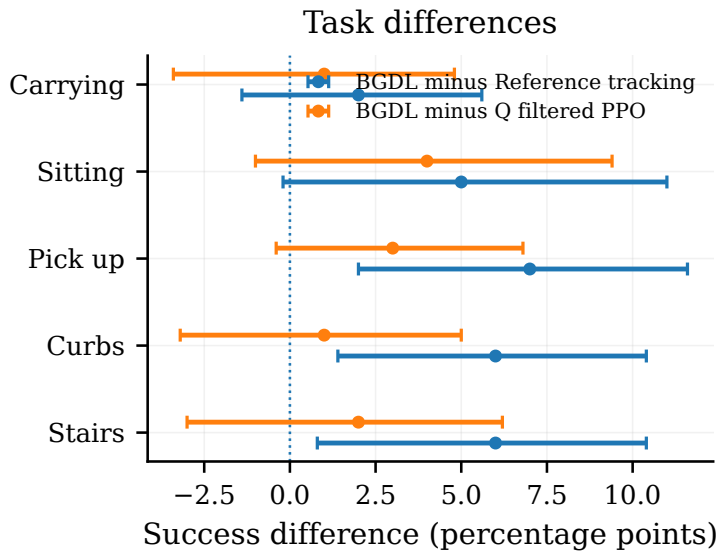

Figure 20: **Task-specific success differences.** Points subtract each baseline mean from the BGDL mean. Intervals independently resample training runs for the two methods within each task and are reported as pointwise bootstrap intervals.

F.8 Training and Execution Cost

Training and inference are timed on the same device and at the same numerical precision, using a fixed batch size and number of parallel environments. GPU execution is synchronized before each timing measurement. We report actor and critic optimization, failure-predictor fitting, peak memory use, and deployed policy inference separately. Deployment timing includes observation preprocessing and action decoding but excludes the failure predictor, which is not part of the runtime policy. Table 15 separates the additional training cost of BGDL from the computation required at deployment. The failure predictor adds computation during training but does not increase the inference path of the deployed actor.

Table 15: **Training and inference cost.** Update and auxiliary entries are reported in milliseconds per optimization block. Inference includes observation preprocessing, the actor, and action decoding. Training uses a batch size of 4,096, with the same device, precision, and timing boundaries for all methods.

Method	Update (ms)	Aux. (ms)	Memory (GiB)	Inference (ms)
Uniform matched	80.2	0.0	17.5	2.13
Q filtered PPO	84.5	8.1	18.2	2.10
BGDL	85.5	7.2	18.4	2.06

F.9 Sensitivity to the Physical Guidance Threshold

The physical threshold ϵ_h determines when the physical-margin gate begins to increase during training. In contrast, the diagnostic cutoff w_0 is used only to partition states after training and is therefore not a training hyperparameter. Table 16 varies ϵ_h while keeping all other training settings fixed. Each threshold is trained independently using the same validation allowance. Physical evaluation criteria, recovery thresholds, and test scenes remain fixed. We report the mean guidance weight together with task outcomes to separate changes in where supervision is applied from simply increasing the overall amount of imitation.

Table 16: **Physical threshold sweep.** BVR denotes violating transitions, while BRR uses the separately defined set of resolved near-boundary recovery events.

ϵ_h	Success (%)	BVR (%)	BRR (%)	Mean w_n
0.02	87.6	13.8	70.2	0.24
0.05	89.8	10.5	76.9	0.31
0.10	91.2	8.1	82.1	0.38
0.20	90.1	7.3	83.0	0.51
0.40	86.7	6.5	84.2	0.66

G Limitations and Scope

BGDL improves how a fixed set of demonstrations is used during policy learning, but it does not remove the limitations of that demonstration set. If the learned policy reaches states that are poorly covered by the available trajectories, increasing or decreasing the imitation weight cannot provide a new expert action. The method can therefore benefit from good coverage of the parts of the task where precise behavior is required. A future extension is to use the same notion of local action restriction to decide where additional demonstrations should be collected.

The failure predictor has a related limitation. It is learned from policy rollouts and therefore reflects the state distribution encountered during training. Its calibration degrades when the policy or environment moves away from that distribution. For this reason, we use the predictor only to adjust demonstration supervision during training. It is not treated as a safety certificate and is not queried by the deployed policy.

The physical margins are also task-dependent approximations of local constraint slack. Foothold, collision, and balance margins capture important sources of failure in the tasks considered here, but they do not describe every aspect of whole-body dynamics. Required interaction distances address a different case, where precision matters even though the desired contact is not unsafe. More complex manipulation or highly dynamic contact sequences may require richer task-specific cues.

Our theoretical analysis is deliberately local. The one-step result assumes a positive expert margin, bounded geometry estimation error, and local smoothness of the estimated margin with respect to the action. These conditions explain how the allowable deviation around a demonstrated action changes with physical slack and action sensitivity, but they do not imply safety over an entire trajectory. Contact mode changes and unmodeled disturbances can also invalidate the local assumptions. Finally, the physical robot evaluation covers a limited number of trials and task conditions. It shows that the learned policies can transfer to the tested Unitree G1 setup without policy fine-tuning, but it should not be interpreted as evidence of reliability across arbitrary hardware states, terrain, objects, or disturbances. Broader physical evaluation and pairing BGDL with an independent execution time constraint mechanism are important directions for future work.