

Action-Aligned Latent Planning for Versatile Humanoid Loco-Manipulation

Yutang Lin¹, Guanya Shi^{†,1,2}, C. Karen Liu^{†,1,3}

One Autonomous Vision-Conditioned Policy, Diverse Real-World Interactions

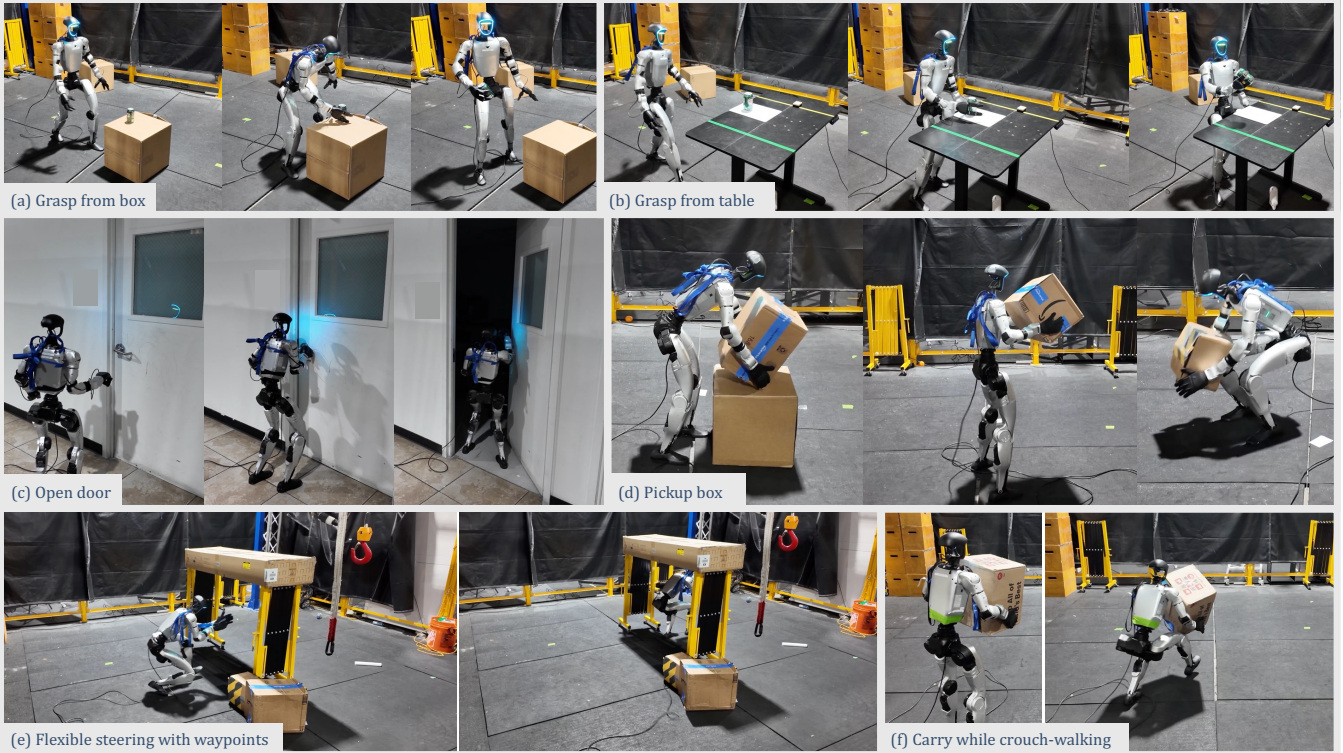


Fig. 1: A single autonomous perceptive latent planner enables diverse real-world interactions, including object grasping in (a,b), door opening in (c), whole-body box pickup in (d), waypoint-conditioned steering in (e), and crouched locomotion while carrying in (f).

Abstract—Learning a single perceptive policy for diverse humanoid loco-manipulation requires unifying high-level task decisions without losing the state-dependent physical responses that make each interaction succeed. This makes the representation between high-level planning and whole-body control critical: kinematic or contact targets provide compact abstractions but do not directly encode interaction-specific control responses, while directly predicting low-level actions requires the planner to model high-frequency closed-loop feedback. We propose FIBER, a framework that uses the executable latent space of a frozen Behavior Foundation Model (BFM) as a shared planning representation for heterogeneous interaction skills. Our Latent Policy Projection (LPP) maps independently acquired task-specific policies into this representation by identifying latent commands whose induced actions match their local control responses, enabling lower-rate planning while retaining the BFM’s high-frequency state feedback. After projection, we aggregate closed-loop latent rollouts from all skills and train a

single perceptive flow planner entirely offline. We further introduce AH-BFM, an adaptive-horizon BFM trained with multi- γ Forward-Backward (FB) learning to support both precise control and temporally extended whole-body behavior. We evaluate FIBER on box pickup, door opening, and object grasping in simulation and on a real Unitree G1 humanoid. Across 128 diverse variations per task in simulation, FIBER achieves 42.0% mean success, more than tripling projected latent replay (11.6%) and substantially outperforming kinematic planning (3.5%). Real-world experiments further demonstrate sim-to-real transfer, skill chaining, and conditional steering through the same unified planner. Project website: <https://fiberaa.github.io/>.

I. INTRODUCTION

Humanoid loco-manipulation requires both deciding *what* interaction to perform from visual and semantic context and determining *how* to realize it through balance, contact, and coordinated whole-body control. Recent work has achieved strong task-specific interaction skills, including object manipulation, articulated-object interaction, and object transport [1, 2], often using task-specific objectives, observations,

¹Amazon FAR.

²Carnegie Mellon University.

³Stanford University.

[†]Amazon FAR Team Co-Lead.

and training procedures. Scaling these successes to a versatile policy requires unifying diverse task-level decisions without losing the physical responses that make each skill succeed.

A common strategy separates task-level decision making from whole-body control. A high-level policy predicts structured targets such as body motion, end-effector poses, or contacts, while a low-level controller realizes them on the robot. However, the same target may require different actions under different contacts, object states, or disturbances, so such representations do not directly capture the state-dependent control responses required for interaction. Predicting low-level actions avoids this abstraction, but offline learning then requires broad coverage of closed-loop corrective responses, which is difficult for contact-rich interaction. This exposes a central challenge for humanoid planning: *how can we abstract low-level control complexity without abstracting away the physical responses that make interaction succeed?*

A learned behavioral representation offers a way to resolve this tension. We build on a Behavior Foundation Model (BFM) that exposes an executable behavioral latent space. Given the current robot state and a latent command, BFM produces low-level actions at high frequency. A latent therefore indexes a state-dependent control policy rather than merely a desired motion, allowing high-level planning to operate over compact behavioral commands while the BFM retains high-frequency closed-loop physical feedback.

Using such an interface for versatile loco-manipulation still requires acquiring the interaction capabilities that the planner should invoke. One approach is to collect demonstrations for the desired interactions through a common data-acquisition pipeline [3, 4], but obtaining broad coverage of contact-rich and corrective behavior is expensive. Another approach is to first train task-specific interaction policies and distill them into a generalist. On-policy multi-expert distillation can improve closed-loop coverage by querying these policies at states visited by the generalist, but requires all policies and their environments to remain available during integration. We instead consider a decoupled setting: each interaction policy can be acquired independently, and only afterward incorporated into the shared behavioral interface.

Under this decoupled setting, different interaction policies need not share their observations, objectives, or task representations. They nevertheless share one quantity that directly describes their physical realization: the actions applied to the robot. A latent command from the frozen BFM is grounded in the same quantity, because it also produces robot actions conditioned on the current robot state. This common action space provides a direct way to relate an independently acquired interaction policy to the BFM latent representation.

Based on this insight, we propose **FIBER**, which represents independently acquired interaction policies, or experts, in the executable latent space of a frozen BFM. At the core of **FIBER** is Latent Policy Projection (LPP), which identifies latent commands by matching the actions induced by the BFM to an expert’s local state-dependent control responses. The resulting projected policy executes through the frozen BFM, preserving high-frequency state feedback

while expressing heterogeneous interaction skills through a common lower-rate representation.

Because each expert is projected independently, the experts need not share an acquisition procedure or remain jointly available during subsequent training. After projection, we roll out the projected policies and collect their latent commands together with the perceptive and proprioceptive observations available to the final policy. These closed-loop latent rollouts are aggregated across skills to train a single perceptive planner entirely offline. At deployment, the planner predicts future latent commands from onboard observations and a high-level steering condition, while the frozen BFM provides the state-dependent whole-body responses required to execute them.

Finally, projection and planning impose different temporal requirements on the latent representation. Matching an expert’s control response favors precise short-horizon behavior, whereas lower-rate planning benefits from stable commands that preserve behavior over longer horizons. We therefore introduce **AH-BFM**, an adaptive-horizon BFM trained with multi- γ Forward-Backward (FB) learning to support both control fidelity and temporally extended behavioral structure.

We evaluate **FIBER** on three heterogeneous interaction tasks—box pickup, door opening, and object grasping—in simulation and on a real Unitree G1. In simulation, LPP largely preserves expert performance on box pickup and door opening near the interaction regime. The unified perceptive planner achieves 42.0% mean success, compared with 11.6% for fixed projected latent replay and 3.5% for kinematic planning. Real-world experiments further demonstrate direct sim-to-real transfer, interaction chaining, and waypoint-conditioned steering through the same planner.

Our contributions are threefold:

- **A shared executable representation for humanoid interaction planning.** We use the latent space of a frozen BFM as a behavioral interface that supports lower-rate high-level planning while retaining state-dependent interaction realization responses.
- **Action-aligned latent policy projection.** We introduce LPP, which maps independently acquired interaction experts into the shared BFM representation by aligning their local control responses in robot action space, decoupling skill acquisition from subsequent integration.
- **Adaptive-horizon BFM and SF-guided behavior reuse.** We introduce multi- γ FB learning and **AH-BFM** that balance precise control with temporally extended behavioral structure, together with Successor-Feature (SF)-guided interaction-preserving behaviors reuse.

II. RELATED WORK

A. Humanoid Controller and Behavioral Foundation Models

Humanoid control has increasingly moved from skill-specific policies toward shared controllers that support broad motor repertoires through a common conditioning interface. Motion-conditioned approaches unify locomotion and whole-body behaviors through dense or sparse motion specifications [5, 6], while recent systems extend unified control

toward language-conditioned behaviors and humanoid loco-manipulation [7, 8]. These works demonstrate that diverse behaviors can be consolidated within a single controller, but the supported behaviors are typically acquired through the controller’s predefined training and conditioning interface. Our work instead considers a complementary setting: integrating interaction behaviors acquired independently of this interface into an already-trained controller.

BFMs expose a learned behavioral latent space as such a shared control interface. BFM-Zero [9] uses FB to learn a promptable humanoid behavioral space, while subsequent behavioral inference methods show that expert behaviors can be recovered by optimizing within a frozen FB latent space [10]. These results establish latent inference as an effective mechanism for behavior reproduction. We build on this capability but study its use as an interface for interaction representation, where the latent sequence must capture temporally varying local control responses rather than only recover a stationary behavior or desired state trajectory. This setting further exposes a trade-off in conventional FB learning: the successor horizon favorable for precise local tracking differs from that required to preserve long-term behavior. Our adaptive-horizon BFM is designed to retain both properties within the same behavioral space.

B. Humanoid Loco-Manipulation and Representation

Humanoid loco-manipulation requires specifying not only the desired motion of the robot, but also how it should respond to physical interaction with objects and the environment. A broad family of methods makes interaction explicit through structured task representations, including contact sequences, end-effector constraints, task-motion objectives, distance fields, and interaction-preserving references [11–17]. Such representations provide effective and often interpretable interfaces for learning diverse skills. However, they specify interaction through variables chosen in advance; the resulting task description does not directly encode the state-dependent control response induced by contact and environment dynamics.

Alternatively, interaction responses can be learned directly through task-specific reinforcement learning (RL) or demonstrations [18–25]. Teleoperation and human-video-based pipelines further broaden the sources from which such behaviors can be acquired [26–32]. These approaches can capture interaction-specific control responses through RL, but the acquired behavior generally remains tied to the specialist policy or downstream training process. In contrast, we decouple interaction acquisition from execution: demonstrated state-action responses are projected into the native latent interface of a frozen behavioral foundation controller, allowing independently acquired interactions to share a common executable substrate.

III. METHODOLOGY

A. Overview

Our goal is to express independently acquired interaction skills in a shared behavioral interface that preserves their

state-dependent physical realization and supports unified perceptive planning. **FIBER** uses the latent space of a frozen BFM as this interface and maps each interaction expert into it by aligning the expert’s local control responses with the actions induced by BFM latent commands.

As illustrated in Fig. 2, **FIBER** proceeds in three stages. First, interaction skills are acquired independently with task-appropriate supervision. Second, LPP maps each expert into the frozen BFM latent space by matching the actions induced by latent commands to the expert’s local control responses. The resulting projected policies are rolled out directly through the shared behavioral interface, after which the original experts are no longer required. Finally, their closed-loop latent rollouts are aggregated to train a single perceptive flow planner entirely offline, conditioned on on-board observations and high-level steering commands.

This latent space must support both precise local interaction and temporally extended whole-body behavior. We therefore introduce **AH-BFM**, an adaptive-horizon BFM trained with multi- γ FB learning. We consider

$$\Gamma = \{\gamma_s, \gamma_l\}, \quad \gamma_s < \gamma_l, \quad (1)$$

where γ_s emphasizes local transition fidelity and γ_l preserves longer-horizon behavioral structure. We condition the forward representation F on γ while sharing the backward representation B , and normalize the corresponding values as

$$\tilde{Q}_\gamma^{\text{FB}} = (1 - \gamma)Q_\gamma^{\text{FB}}. \quad (2)$$

The shared actor adaptively aggregates the horizon-specific objectives,

$$J_\pi = \mathbb{E} \left[\tau \log \sum_{\gamma \in \Gamma} p(\gamma) \exp \left(\frac{\tilde{Q}_\gamma^{\text{FB}}(s, \pi_{\text{BFM}}(s, z), z)}{\tau} \right) \right], \quad (3)$$

with gradient weights

$$w_\gamma \propto p(\gamma) \exp \left(\frac{\tilde{Q}_\gamma^{\text{FB}}(s, \pi_{\text{BFM}}(s, z), z)}{\tau} \right). \quad (4)$$

Here, $p(\gamma)$ defines a prior over horizons and τ controls the aggregation sharpness. Because the weights depend on the current state and latent command, the shared actor can combine short- and long-horizon supervision within the same behavioral space.

B. Heterogeneous Skill Acquisition

FIBER allows interaction skills to be acquired with task-appropriate supervision, observations, and training objectives. We consider three complementary acquisition routes.

a) Motion-based acquisition: For interactions with suitable humanoid–object motion references, such as grasping [33] and object pickup [29], we train task-specific experts by combining motion imitation with interaction objectives.

b) Reward-shaping-based acquisition: Some interactions admit a natural decomposition into successive stages and can therefore be acquired directly through staged reward shaping. For example, opening an articulated door [2] can

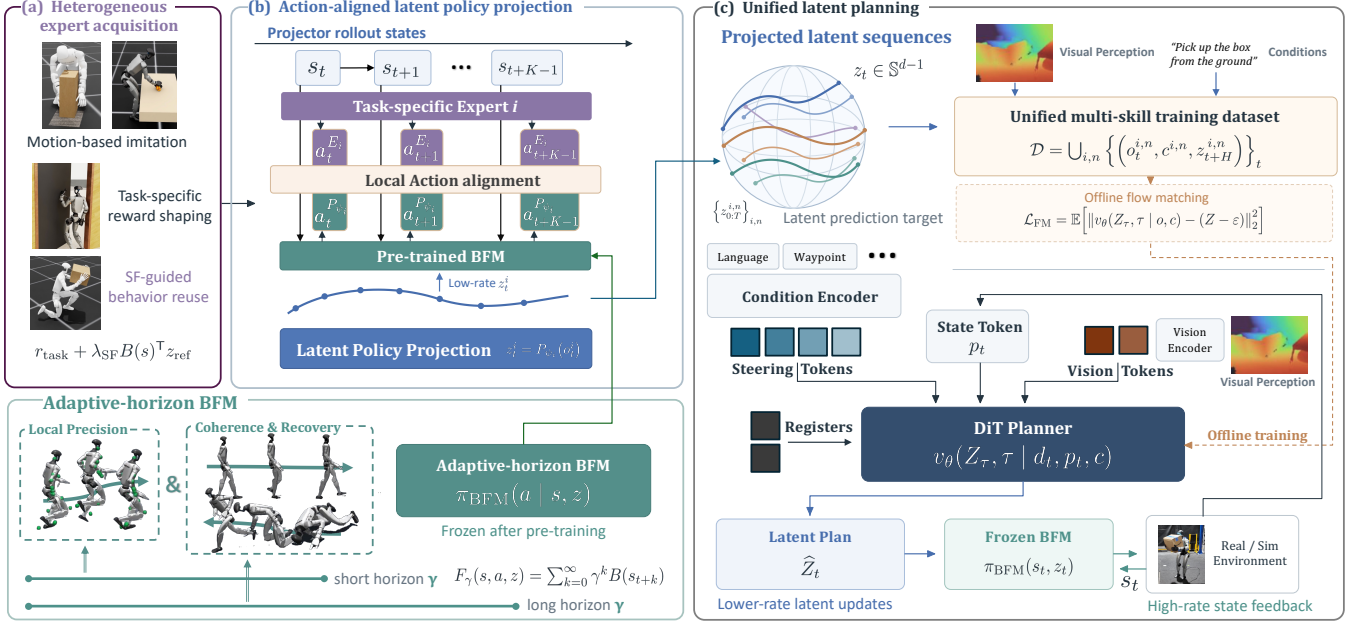


Fig. 2: **Overview of FIBER.** (a) Heterogeneous interaction experts are acquired through motion imitation, reward shaping, or successor-feature-guided behavior reuse. (b) Action-aligned latent policy projection maps their short-horizon control responses into a frozen adaptive-horizon BFM; closed-loop rollouts produce executable latent trajectories. (c) These trajectories train a unified perceptive planner entirely offline. Language commands or waypoint conditions are encoded as steering tokens, and a DiT planner predicts future latent trajectories. During deployment, the planner replans from onboard observations while the frozen BFM provides high-frequency feedback.

be trained by progressively rewarding approach, contact establishment, handle manipulation, and door opening. This provides a direct route to task-specific experts without requiring motion references, while allowing each stage to use task-appropriate observations and rewards.

c) SF-guided behavior reuse: The pretrained BFM provides an additional route for extending acquired interactions with whole-body capabilities already represented in its latent space. For an interaction I , such as carrying box, we learn a latent adapter A_{ψ_I} while keeping the BFM frozen:

$$\hat{z}_t = A_{\psi_I}(s_t, z^N), \quad a_t = \pi_{\text{BFM}}(s_t, \hat{z}_t), \quad (5)$$

where z^N specifies the native behavior to be incorporated.

We optimize the adapter using the interaction reward together with a successor-feature-guided preservation term,

$$r_t = r_t^I + \lambda_{\text{SF}} B(s_t)^\top z^N. \quad (6)$$

For a fixed z^N ,

$$\mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t B(s_t)^\top z^N \right] = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t B(s_t) \right]^\top z^N, \quad (7)$$

so the second term encourages the adapted rollout to preserve the behavior specified by z^N while r_t^I maintains the interaction. This enables acquired interactions to be extended with native skills, such as running or kneeling while carrying.

Motion imitation, task-specific reward learning, and successor-feature-guided behavior reuse each produce task-specific experts, all of which are subsequently projected into the shared BFM latent space through LPP.

C. Action-Aligned Latent Policy Projection

Externally acquired experts must be converted into a representation suitable for offline planner training. Kinematic trajectories provide compact behavioral targets, but do not uniquely determine the control responses required to realize an interaction. Expert actions provide this missing physical information, yet directly modeling high-frequency actions requires covering a broad state-dependent control distribution. We instead use the frozen BFM to map expert behavior into a lower-rate latent representation that preserves both its executable response and native behavioral semantics.

For expert π_i^E , we learn a local projector P_{ψ_i} that maps the expert observation at update time t_k to a latent,

$$z_k = P_{\psi_i}(o_{t_k}). \quad (8)$$

Between consecutive updates, we interpolate from the previous latent z_{k-1} to z_k using SLERP to obtain a latent command at each control step,

$$z_{t_k+h} = \text{SLERP}(z_{k-1}, z_k, \alpha_h), \quad h = 0, \dots, H_P - 1. \quad (9)$$

We identify the latent command primarily through its induced actions. Given expert actions

$$a_t^{E_i} = \pi_i^E(o_{t_i}^E), \quad (10)$$

we define the short-horizon action-alignment objective

$$\mathcal{L}_{\text{act}} = \mathbb{E}_{P_{\psi_i}} \left[\frac{1}{H_P} \sum_{h=0}^{H_P-1} \left\| \pi_{\text{BFM}}(s_{t_k+h}, z_{t_k+h}) - a_{t_k+h}^{E_i} \right\|_2^2 \right]. \quad (11)$$

Here, P_{ψ_i} denotes the on-policy state distribution induced

TABLE I: Simulation success rates (%) across heterogeneous loco-manipulation tasks. “w/ appr.” includes the approaching phase, whereas “w/o appr.” initializes the humanoid near the interaction state. [†] denotes a task-specific policy, and * denotes access to privileged object states instead of egocentric observations. Mean is computed using the “w/ appr.” results.

Method	Pickup Box		Open Door		Grasp Object	Mean
	w/ appr.	w/o appr.	w/ appr.	w/o appr.		
Task-specific policies						
Expert Policy [†]	100.0±0.0	90.8±0.5	89.3±0.2	97.9±0.8	97.0±0.8	95.4
LPP w/ SONIC [34] [†]	0.0±0.0	0.0±0.0	83.1±0.7	88.1±0.6	5.1±0.7	29.4
LPP w/ Fixed- γ BFM [†]	75.4±0.2	80.2±1.3	61.8±1.9	72.4±0.7	69.3±1.2	68.8
LPP w/ AH-BFM [†]	81.9±0.9	87.4±2.4	83.5±1.8	97.3±0.4	52.9±2.1	72.8
Reference-prompted execution						
SONIC [34]	0.0±0.0	0.0±0.0	0.4±0.0	3.1±0.0	12.1±0.0	4.2
FALCON [21]	0.0±0.0	0.0±0.0	0.4±0.0	2.7±0.0	11.9±0.3	4.1
$B(s)$ Tracking	0.0±0.0	0.0±0.0	0.4±0.0	5.6±0.5	13.9±0.5	4.8
LPP Replay	11.2±0.8	17.7±1.2	12.2±0.9	49.5±0.6	11.3±1.1	11.6
Perceptive planning						
OmniContact* [17]	29.9±0.2	49.6±0.0	—	—	—	—
LessMimic [14]	38.3±0.0	28.5±0.0	—	—	—	—
Action Planner	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0
Kinematic Planner	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	10.6±0.5	3.5
Ours – FIBER	42.1±0.6	44.1±1.0	53.5±4.8	51.2±1.0	30.3±2.3	42.0

by the current projector executed through the frozen BFM. This directly constrains the projected latent by matching the expert’s responses, rather than by a predefined kinematic or contact representation target.

Action alignment alone, however, can leave the latent underdetermined: different commands may induce similar local actions while corresponding to different behavioral directions in the BFM space. Recall that the BFM latent z acts as a policy index through the forward-backward objective $F(s, a)^\top z$, while $B(s)$ provides the corresponding state-side behavioral representation. We therefore anchor each projected command to the native semantic representation of its local state,

$$\mathcal{L}_{\text{sem}} = \mathbb{E} \left[\|z_k - B(s_{t_k})\|_2^2 \right]. \quad (12)$$

The complete projection objective is

$$\mathcal{L}_{\text{proj}} = \mathcal{L}_{\text{act}} + \lambda_{\text{sem}} \mathcal{L}_{\text{sem}}. \quad (13)$$

The action term determines the latent according to the expert’s required control response, while the semantic term selects solutions consistent with the native behavioral geometry of the BFM. This reduces action-equivalent latent ambiguity and yields more coherent latent trajectories for downstream planning. To obtain a robust projection beyond nominal expert rollouts, we train each P_{ψ_i} independently using single-skill on-policy data aggregation [35].

After projection, the original expert is no longer required. We roll out P_{ψ_i} under randomized initial states and environment configurations and collect

$$\mathcal{D}_i^Z = \{(d_t, p_t, u_t, z_t)\}, \quad (14)$$

where d_t and p_t denote the perceptive and proprioceptive observations available to the downstream planner, and u_t denotes the corresponding high-level task condition. These latent rollouts provide executable and semantically regularized targets that can be combined across independently acquired skills for offline latent planning.

D. Unified Latent Flow Planning

Given the collected latent rollouts, we train a single perceptive planner to generate future behavioral commands entirely offline. At each planning step, the policy is conditioned on onboard depth d_t , proprioception p_t , and a compact steering token c_t :

$$c_t = E_{\text{steer}}(u_t). \quad (15)$$

The steering input u_t specifies the desired high-level behavior and can be instantiated by different conditioning modalities, such as a language command or a navigation waypoint. This separates the representation of task intent from the perceptive latent planner itself.

The planner predicts a horizon of future BFM commands

$$x_1 = \text{vec}(z_t, z_{t+1}, \dots, z_{t+H-1}) \quad (16)$$

using conditional flow matching. Given Gaussian noise $x_0 \sim \mathcal{N}(0, I)$ and flow time $\tau \sim \mathcal{U}(0, 1)$, we construct the linear interpolation $x_\tau = (1 - \tau)x_0 + \tau x_1$. Then a conditional network predicts the clean latent, and is trained with

$$\mathcal{L}_{\text{flow}} = \mathbb{E} \left[\|\hat{x}_1 - x_1\|_2^2 \right]. \quad (17)$$

At deployment, the planner repeatedly observes the current depth and proprioceptive state, constructs the steering

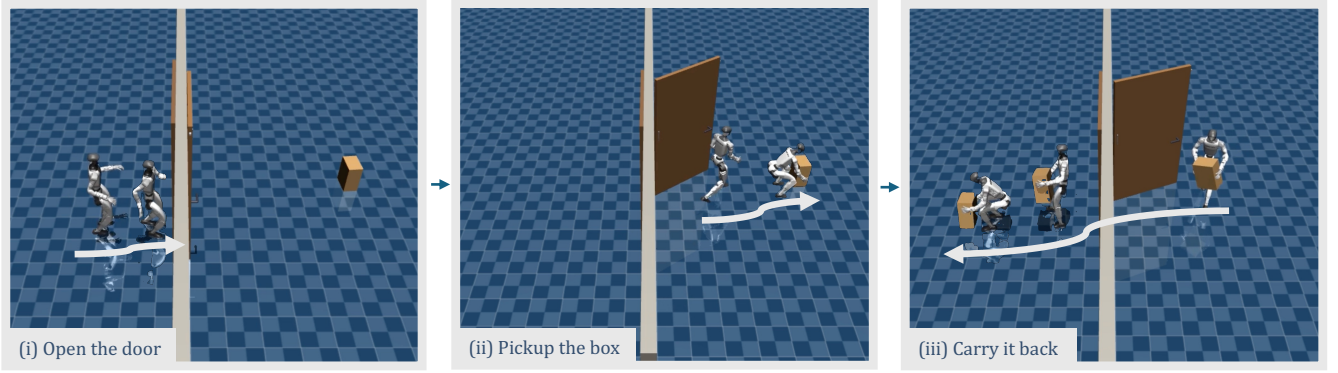


Fig. 3: **Skill chaining with FIBER.** In a single rollout, the policy sequentially opens the door in (i), picks up the box in (ii), and carries it back through the doorway in (iii), demonstrating closed-loop chaining of multiple interaction skills under changing high-level commands.

token from the desired high-level condition, and generates a future latent trajectory. The leading latent commands are executed through the frozen BFM, after which the planner replans from the newly observed state. This receding-horizon execution allows the same policy to continuously adapt its whole-body behavior to the perceived scene while following different steering conditions.

Through this pipeline, skills originating from heterogeneous acquisition procedures are converted into a common executable latent representation and subsequently learned by a single offline perceptive planner.

IV. EXPERIMENTS

We evaluate **FIBER** through simulation and real-world experiments to answer four questions:

Q1: How faithfully can heterogeneous interaction experts be projected into the shared latent space of a frozen BFM?

Q2: Can these projected skills be jointly learned by a single planner for versatile humanoid loco-manipulation?

Q3: Can the unified planner chain projected skills and flexibly steer their execution through various conditions?

Q4: Which components of **FIBER** are critical for reliable latent projection and downstream planning?

For each projected skill, we collect 20K episodes, and merge them across tasks for **FIBER** training. Language conditions are assigned according to the underlying task, while waypoint conditions are labeled from the robot trajectories in the collected rollouts.

A. Simulation Evaluation

We evaluate in IsaacSim, with **128 variations per task**: box size and initial placement for pickup, door geometry for opening, and object identity for grasping. We do not report on-policy multi-expert distillation because the experts use different environments, observation spaces, and codebases, and therefore do not share the online rollout interface required for joint distillation.

a) Action-aligned latent projection: To answer **Q1**, we compare each task-specific expert with its LPP projector in Tab. I. LPP largely preserves expert performance on box pickup and door opening, while fine-grained grasping is more challenging, dropping from 97.0% to 52.9%.

We further compare representations using targets derived from the same successful expert trajectories. Reference-tracking baselines, including the BFM’s own backward representation $B(s)$, perform poorly, whereas action-aligned projected replay achieves substantially higher success on box pickup and door opening. Notably, **FALCON** [21], despite learning force-adaptive end-effector tracking under randomized external forces, achieves only 0.4%/2.7% on door opening. These results show that kinematic, behavioral, or generic force-adaptive tracking alone does not recover the task-specific control responses required for interaction, motivating action-aligned latent projection.

b) Unified perceptive planning: To answer **Q2**, we compare replaying LPP-produced latent trajectories with the unified perceptive planner. **FIBER** improves mean success from 11.6% to 42.0%, showing that high-frequency BFM feedback alone is insufficient and that task-level latent replanning remains important. Under the same perceptive planning setup, the Action Planner fails on all tasks, while the Kinematic Planner reaches only 3.5% mean success. Directly planning low-level actions removes high-frequency state feedback between planner updates, whereas predicting kinematic targets does not specify the interaction-specific control response, which leads to failure on contact-rich tasks; **FIBER** retains both through latent planning over BFM.

c) Skill chaining and conditional steering: To answer **Q3**, we further test whether the unified planner can compose projected skills under changing high-level commands. As shown in Fig. 3, the same policy sequentially opens the door, picks up the box, and carries it back through the doorway within a single rollout, demonstrating closed-loop chaining of heterogeneous interaction skills.

B. Real-World Evaluation

We further deploy **FIBER** on a Unitree G1 equipped with an onboard RealSense depth camera to evaluate real-world transfer of the unified planner (**Q2**) and interaction chaining and steering (**Q3**). Both the planner and **AH-BFM** run on a single NVIDIA RTX PRO 5000 Blackwell laptop. With TensorRT, the planner runs at 5 Hz using 10 Heun steps and Real Time Chunking (RTC) (~20 ms per inference), while

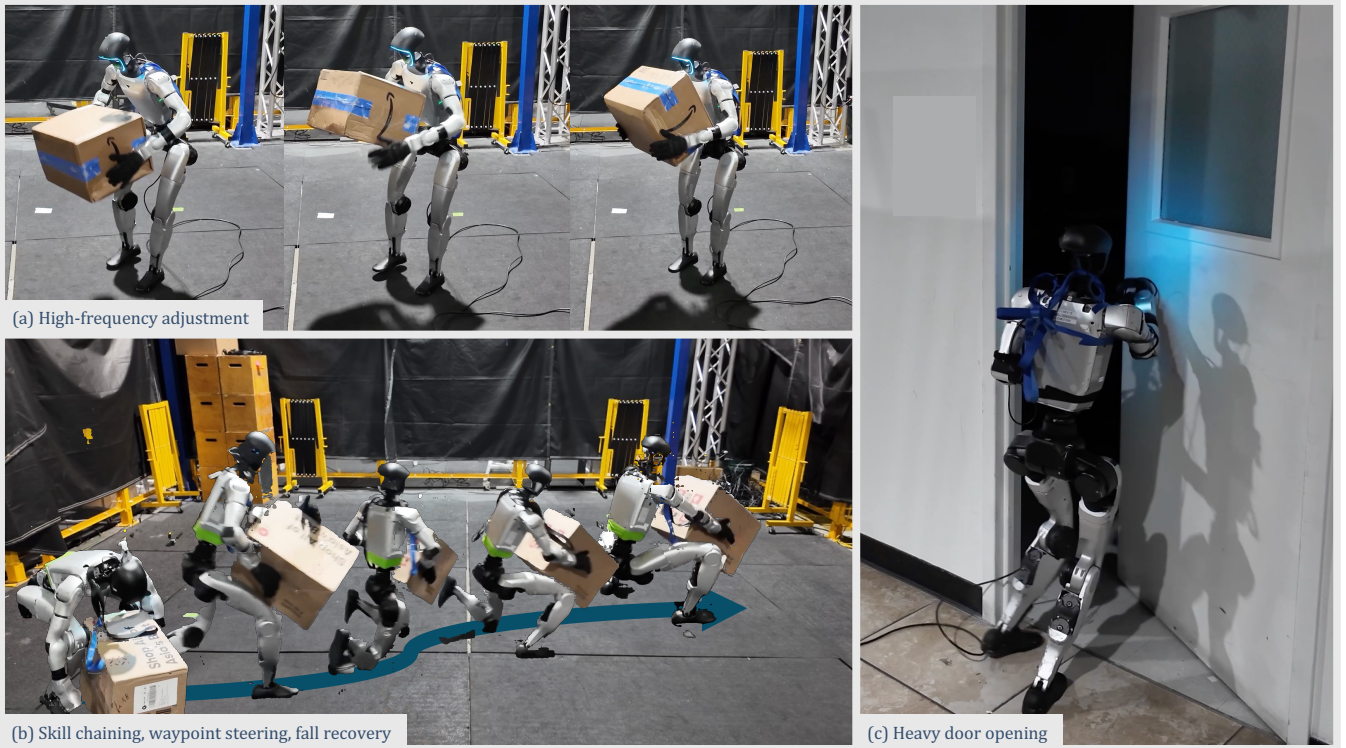


Fig. 4: **Qualitative real-world behaviors of FIBER.** (a) During box carrying, the robot reacts to an unstable grasp by briefly lifting the box and repositioning its hands underneath for more secure support, illustrating high-frequency closed-loop adjustment. (b) After picking up the box, the same policy follows waypoint commands while carrying it; after stumbling, the robot stabilizes on its knees and recovers before continuing, demonstrating interaction-preserving skill chaining, steering, and recovery. (c) For a heavy door, the robot leans its whole body into the interaction to generate sufficient pushing force, showing adaptive whole-body realization of contact-rich tasks.

AH-BFM runs at 50 Hz (~ 1 ms per forward pass).

Tab. II reports success rates for box pickup, door opening, and object grasping. All experts, latent projectors, and the unified planner are trained entirely in simulation. **FIBER** succeeds in 6/10 box-pickup trials and 4/10 door-opening trials, demonstrating direct sim-to-real transfer of the latent planning interface. Object grasping is more challenging (2/10), where inaccurate depth estimates for the small object compound the fine-grained control difficulty. Beyond task success, **Fig. 4** highlights the key advantage of our representation: planned latents remain grounded in closed-loop physical responses. When the box becomes unstable, the robot repositions its hands underneath for support in (a); when opening a heavy door, it leans its whole body into the interaction to generate sufficient force in (c).

To answer **Q3**, we further evaluate post-pickup behavior chaining and waypoint steering. After picking up a box, the same planner follows root-waypoint commands while maintaining the grasp; when disturbed during carrying, it can stabilize on its knees, recover, and continue execution, as shown in **Fig. 4** (b). These results show that the unified planner can chain interaction and locomotion behaviors while remaining steerable through high-level waypoint commands.

C. Ablation and Analysis

To answer **Q4**, we analyze the two components underlying our latent planning interface: the adaptive-horizon BFM and

TABLE II: Real-world success rates of the unified **FIBER** planner on the Unitree G1.

Method	Pickup Box	Open Door	Grasp Object	Mean
FIBER	6/10	4/10	2/10	40%

TABLE III: Ablation of multi-horizon BFM tracking and behavior embedding performance on 100STYLE [36].

Method	Joint-MAE $\downarrow$	BE-EMD $\downarrow$			
		0.2 s	0.5 s	1 s	2 s
SONIC [34]	0.1315	—	—	—	—
BFM $\gamma_{0.8}$	0.1107	0.038	0.200	0.495	0.792
BFM $\gamma_{0.98}$	0.1327	0.048	0.213	0.451	0.601
AH-BFM	0.1020	0.049	0.195	0.422	0.590

the action-aligned latent representation.

a) Effect of the adaptive-horizon BFM: We first evaluate how the BFM temporal horizon affects both behavior representation and downstream projection.

For behavior representation, we evaluate **AH-BFM** using two complementary metrics. Joint-MAE measures instantaneous motion-tracking accuracy, while BE-EMD measures how well a fixed latent command preserves a reference behavior over increasing embedding temporal horizons: for each reference clip, we infer a single latent through LPP,

roll out the BFM under this fixed command, and measure its discrepancy from the reference.

As shown in **Tab. III**, the short-horizon model $\gamma = 0.8$ achieves the lowest BE-EMD at 0.2s, whereas $\gamma = 0.98$ better preserves behavior over longer horizons. **AH-BFM** achieves the lowest Joint-MAE and the lowest BE-EMD at 0.5, 1, and 2 s. Adaptive multi- γ learning therefore provides a better trade-off between precise local control and temporally extended behavioral structure than either fixed horizon alone.

This advantage also transfers to interaction projection: in **Tab. I**, replacing the fixed- γ BFM ($\gamma = 0.98$) with **AH-BFM** increases mean LPP success from 68.8% to 72.8%, with substantial gains on box pickup and door opening.

b) *Effect of action-aligned latent representation*: The reference-prompted results in **Tab. I** isolate the importance of action alignment. $B(s)$ tracking and projected replay are derived from the same successful expert trajectories, but the former identifies behavior from the expert state trajectory whereas LPP identifies latent commands from the expert’s action responses. Their large performance difference on pickup box and door opening shows that behavioral-state matching alone does not retain interaction-critical feedback. At the other extreme, the Action Planner fails in all tasks, showing that low-rate action planning cannot replace the high-frequency state feedback provided by the frozen BFM.

The gap between the feedback-conditioned LPP projector and fixed projected replay further shows that projected behaviors cannot generally be reduced to open-loop latent trajectories. Conversely, the improvement from projected replay to **FIBER** shows that a unified perceptive planner can recover task-level latent feedback from onboard observations. Together, these comparisons separate the roles of action-aligned representation, latent feedback, and perceptive replanning in **FIBER**.

V. CONCLUSIONS AND LIMITATIONS

We presented **FIBER**, an action-aligned latent planning framework for versatile humanoid loco-manipulation. LPP projects heterogeneous interaction experts into a frozen BFM by matching their short-horizon control responses, producing executable latent trajectories for unified offline perceptive planning. Adaptive multi- γ FB learning supports both precise control and temporally extended behavior, while successor-feature-guided learning enables interaction-preserving behavior reuse. Our simulation and real-world experiments demonstrate effective expert projection and perception-conditioned execution across diverse interaction tasks.

While our approach enables versatile humanoid loco-manipulation through action-aligned latent planning, extending the underlying BFM toward more dynamic whole-body behaviors, such as stair traversal, remains worth exploring. Moreover, incorporating richer visual and tactile feedback together with online latent adaptation could support interaction under more diverse contact conditions and facilitate the continual integration of new skills. We hope our work provides insights into scalable humanoid skill acquisition and opens new directions for unified whole-body planning.

REFERENCES

- [1] T. He, Z. Wang, H. Xue, Q. Ben, Z. Luo, W. Xiao, Y. Yuan, X. Da, F. Castañeda, S. Sastry, C. Liu, G. Shi, L. Fan, and Y. Zhu, “Viral: Visual sim-to-real at scale for humanoid loco-manipulation,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026, pp. 13 430–13 441.
- [2] H. Xue, T. He, Z. Wang, Q. Ben, W. Xiao, Z. Luo, X. Da, F. Castañeda, G. Shi, S. Sastry, L. J. Fan, and Y. Zhu, “Opening the sim-to-real door for humanoid pixel-to-action policy transfer,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.
- [3] Z. Li, Z. Zhang, Y. Wei, W. Zhang, X. Yuan, P. Zhi, G. Li, X. Guo, F. Gao, J. Yang, and S. Zhang, “ ω -0: A latent predictive world action model for concurrent humanoid loco-manipulation,” 2026.
- [4] J. Zheng, T. Ma, Y. Fan, Z. Wang, S. Yang, and J. Liang, “MotionWAM: Towards foundation world action models for real-time humanoid loco-manipulation,” 2026.
- [5] C. Tessler, Y. Guo, O. Nabati, G. Chechik, and X. B. Peng, “Masked-mimic: Unified physics-based character control through masked motion inpainting,” *ACM Transactions on Graphics (TOG)*, vol. 43, no. 6, pp. 1–21, 2024.
- [6] Q. Liao, T. E. Truong, X. Huang, Y. Gao, G. Tevet, K. Sreenath, and C. K. Liu, “Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion,” *Science Robotics*, vol. 11, no. 117, p. eadx8924, 2026.
- [7] H. Jiang, J. Chen, Q. Bu, L. Chen, M. Shi, Y. Zhang, D. Li, C. Suo, C. Wang, Z. Peng, and H. Li, “Wholebodyvla: Towards unified latent vla for whole-body loco-manipulation control,” in *International Conference on Learning Representations (ICLR)*, 2026.
- [8] S. Wei, H. Jing, B. Li, Z. Zhao, J. Mao, Z. Ni, S. He, S. Zang, X. Liu, K. Kang, J. Liu, W. Yuan, M. Pavone, D. Huang, and Y. Wang, “ ψ_0 : An open foundation model towards universal humanoid loco-manipulation,” in *Robotics: Science and Systems (RSS)*, 2026.
- [9] Y. Li, Z. Luo, T. Zhang, C. Dai, A. Kanervisto, A. Tirinzoni, H. Weng, K. Kitani, M. Guzek, A. Touati, A. Lazaric, M. Pirota, and G. Shi, “BFM-Zero: A promptable behavioral foundation model for humanoid control using unsupervised reinforcement learning,” in *International Conference on Learning Representations (ICLR)*, 2026.
- [10] M. Pirota, A. Tirinzoni, A. Touati, A. Lazaric, and Y. Ollivier, “Fast imitation via behavior foundation models,” in *International Conference on Learning Representations (ICLR)*, 2024.
- [11] C. Zhang, W. Xiao, T. He, and G. Shi, “Wococo: Learning whole-body humanoid control with sequential contacts,” in *Conference on Robot Learning (CoRL)*. PMLR, 2024, pp. 455–472.
- [12] Y. Li, Y. Zhang, W. Xiao, C. Pan, H. Weng, G. He, T. He, and G. Shi, “Hold my beer: Learning gentle humanoid locomotion and end-effector stabilization control,” in *Conference on Robot Learning (CoRL)*, ser. Proceedings of Machine Learning Research, vol. 305, 2025, pp. 4506–4523.
- [13] J. Ren, J. Long, T. Huang, H. Wang, Z. Wang, F. Jia, W. Zhang, J. Wang, P. Luo, and J. Pang, “Humanoid goalkeeper: Learning from position conditioned task-motion constraints,” 2025.
- [14] Y. Lin, J. Cui, Y. Li, B. Jia, Y. Zhu, and S. Huang, “LessMimic: Long-horizon humanoid interaction with unified distance field representations,” 2026.
- [15] L. Yang, X. Huang, Z. Wu, A. Kanazawa, P. Abbeel, C. Sferazza, C. K. Liu, R. Duan, and G. Shi, “Omniretarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction,” in *International Conference on Robotics and Automation (ICRA)*, 2026.
- [16] S. Chen, S. Zhao, Z. Wu, J. Li, G. Shi, and C. K. Liu, “Scenebot: Contact-prompted general humanoid whole body tracking with scene-interaction,” 2026.
- [17] R. Yu, X. Lin, J. Ma, Y. Wang, K. Luo, J. Ji, H. Wang, W. Wang, R. Zhang, P. Tan, T. Wu, R. Dai, Q. Chen, and L. Han, “Omniconact: Chaining meta-skills via contact flow for generalizable humanoid loco-manipulation,” 2026.
- [18] Y. Wang, J. Lin, A. Zeng, Z. Luo, J. Zhang, and L. Zhang, “Physshoi: Physics-based imitation of dynamic human-object interaction,” 2023.
- [19] Y. Wang, Q. Zhao, R. Yu, H. W. Tsui, A. Zeng, J. Lin, Z. Luo, J. Yu, X. Li, Q. Chen, J. Zhang, L. Zhang, and P. Tan, “Skillmimic: Learning basketball interaction skills from demonstrations,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 17 540–17 549.

- [20] R. Yu, Y. Wang, Q. Zhao, H. W. Tsui, J. Wang, P. Tan, and Q. Chen, "Skillmimic-v2: Learning robust and generalizable interaction skills from sparse and noisy demonstrations," in *ACM SIGGRAPH Conference Papers*, 2025, pp. 1–11.
- [21] Y. Zhang, Y. Yuan, P. Gurunath, I. Gupta, S. Omidshafiei, A.-a. Aghamohammadi, M. Vazquez-Chanlatte, L. Pedersen, T. He, and G. Shi, "Falcon: Learning force-adaptive humanoid loco-manipulation," in *Learning for Dynamics and Control (LADC)*, 2026, pp. 265–281.
- [22] Y. Du, Y. Li, B. Jia, Y. Lin, P. Zhou, W. Liang, Y. Yang, and S. Huang, "Learning human-humanoid coordination for collaborative object carrying," in *International Conference on Robotics and Automation (ICRA)*, 2026.
- [23] Y. Wang, Q. Zhao, Y. F. Lau, R. Yu, H. W. Tsui, Q. Chen, J. Wang, J. Pang, and P. Tan, "Humanx: Toward agile and generalizable humanoid interaction skills from human videos," 2026.
- [24] D. Li, X. Chen, Q. Wu, B. Chen, S. Wu, H. Wu, G. Zhang, L. Li, M. Zhou, D. Xiang, J. Ma, Q. Zhang, and R. Xu, "Haic: Humanoid agile object interaction control via dynamics-aware world model," in *Robotics: Science and Systems (RSS)*, 2026.
- [25] Z. Wang, Z. Wu, P. Abbeel, R. Duan, J. Malik, C. Sferrazza, C. K. Liu, G. Shi, and A. Kanazawa, "Counterfactual video generation enables scalable humanoid loco-manipulation," 2026.
- [26] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi, "OmniH2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning," in *Conference on Robot Learning (CoRL)*, 2024, pp. 1516–1540.
- [27] Y. Ze, Z. Chen, J. P. Araújo, Z. Cao, X. B. Peng, J. Wu, and C. K. Liu, "Twist: Teleoperated whole-body imitation system," in *Conference on Robot Learning (CoRL)*, 2025, pp. 2143–2154.
- [28] Y. Li, Y. Lin, J. Cui, T. Liu, W. Liang, Y. Zhu, and S. Huang, "Clone: Closed-loop whole-body humanoid teleoperation for long-horizon tasks," in *Conference on Robot Learning (CoRL)*, ser. Proceedings of Machine Learning Research, vol. 305, 2025, pp. 4493–4505.
- [29] H. Weng, Y. Li, N. Sobanbabu, Z. Wang, Z. Luo, T. He, D. Ramanan, and G. Shi, "Hdmi: Learning interactive humanoid whole-body control from human videos," 2025.
- [30] R. Nai, B. Zheng, J. Zhao, H. Zhu, S. Dai, Z. Chen, Y. Hu, Y. Hu, T. Zhang, C. Wen, and Y. Gao, "Humanoid manipulation interface: Humanoid whole-body manipulation from robot-free demonstrations," 2026.
- [31] M. Shi, S. Peng, J. Chen, H. Jiang, T. Li, P. Luo, D. Huang, H. Li, and L. Chen, "Unlocking in-the-wild loco-manipulation with robot-free egocentric demonstration," in *Robotics: Science and Systems (RSS)*, 2026.
- [32] T. Wu, X. Kong, Y. Chen, Q. Yu, H. Ye, J. Li, Y. Wang, and H. Dong, "Sugar: A scalable human-video-driven generalizable humanoid loco-manipulation learning framework," 2026.
- [33] T. Xie, H. Zhang, J. Park, Z. Wang, B. Wen, J. Li, X. Li, Q. Ben, H. Weng, Y. Ye, D. Minor, T. Wang, C. Jiang, S. Fidler, J. Kautz, L. Fan, Y. Zhu, Z. Luo, U. Iqbal, and Y. Yuan, "GRAIL: Generating humanoid loco-manipulation from 3D assets and video priors," 2026.
- [34] Z. Luo, Y. Yuan, T. Wang, C. Li, F. Castañeda, S. Chen, Z.-A. Cao, J. Li, D. Minor, Q. Ben, J. Park, D. Sami, Z. Wang, X. Da, R. Ding, C. Hogg, L. Song, E. Lim, E. Jeong, T. He, H. Xue, W. Xiao, S. Yuen, J. Kautz, Y. Chang, U. Iqbal, L. J. Fan, and Y. Zhu, "SONIC: Supersizing motion tracking for natural humanoid whole-body control," *Science Robotics*, vol. 11, no. 117, p. eaed4592, 2026.
- [35] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, ser. Proceedings of Machine Learning Research, vol. 15, 2011, pp. 627–635.
- [36] I. Mason, S. Starke, and T. Komura, "Real-time style modelling of human locomotion via feature-wise transformations and local motion phases," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 5, no. 1, may 2022.

APPENDIX I

ADAPTIVE-HORIZON BEHAVIORAL FOUNDATION MODEL

This appendix follows the order of the methodology in the main paper. We first detail the adaptive-horizon BFM that defines the shared executable space, then the three routes for skill acquisition, action-aligned latent policy projection, and finally unified offline latent planning.

The same BFM latent space serves two roles in **FIBER**: it is the executable interface used to project interaction experts and the prediction space of the unified planner. Projection favors locally precise responses, while latent planning benefits from commands with longer temporal coherence. The adaptive-horizon BFM is designed to support both requirements within one shared behavioral space.

A. Executable Behavioral Interface and BFM Observations

The body controller commands the 29 actuated joints of the Unitree G1 at 50 Hz, with control interval $\Delta t = 0.02$ s. Tasks using Dex3 hands additionally predict 14 hand-joint targets, seven per hand. These hand targets are produced by a separate branch and are not part of the BFM body action.

Let d_z denote the BFM latent dimension. Executable body commands lie on

$$\mathcal{Z} = \{z \in \mathbb{R}^{d_z} : \|z\|_2 = \sqrt{d_z}\}, \quad \Pi_{\mathcal{Z}}(v) = \sqrt{d_z} \frac{v}{\|v\|_2}, \quad (18)$$

with a fixed nonzero reference vector used only for the degenerate zero-norm case. We use $d_z = 256$. The fixed-radius parameterization keeps the average squared coordinate magnitude independent of the latent dimension.

The BFM actor consumes the current robot state together with causal history. Its current proprioceptive state is

$$s_t^{\text{prop}} = [q_t - q^{\text{default}}; \dot{q}_t; g_t^{\text{body}}; 0.25 \omega_t^{\text{body}}] \in \mathbb{R}^{64}, \quad (19)$$

where g_t^{body} is the gravity direction and ω_t^{body} is the body angular velocity. The actor also receives the previous scaled body action and nine causal proprioceptive/action history frames. Each history frame contains 93 entries: 29 action entries, three angular velocities, 29 joint positions, 29 joint velocities, and three gravity entries. The actor therefore receives $64 + 29 + 9 \times 93 = 930$ proprioceptive/action coordinates before the latent command is appended.

The backward representation additionally receives 463 dimensions of heading-frame whole-body kinematics during BFM training and simulation-based projection. These privileged quantities are not consumed by the deployed BFM actor. Domain-randomization variables and external forces may likewise enter training-time representations and critics without entering the deployed policy.

B. Multi-Horizon Forward-Backward Representation

For a latent-conditioned policy, the forward representation admits the successor-feature interpretation

$$F_{\gamma}(s_t, a_t, z) \approx \mathbb{E} \left[\sum_{h=0}^{\infty} \gamma^h B(s_{t+h+1}) \mid s_t, a_t, \pi_{\text{BFM}}(\cdot, z) \right]. \quad (20)$$

The corresponding FB value is $Q_{\gamma}^{\text{FB}}(s, a, z) = F_{\gamma}(s, a, z)^{\top} z$. We condition the forward representation on γ while sharing both the backward map B and the executable actor across horizons. For every replay transition, a horizon is sampled from the configured set Γ , and the same γ is used in the online prediction and bootstrapped target. The output of B is projected to $\mathcal{Z}$.

For a minibatch of n transitions (s_i, a_i, s_i^+, d_i) , with termination indicator d_i , define

$$M_{ij} = F_{\gamma_i}(s_i, a_i, z_i)^{\top} B(s_j^+), \quad (21)$$

$$R_{ij} = M_{ij} - \gamma_i(1 - d_i)\bar{M}_{ij}, \quad (22)$$

where $\bar{M}_{ij}$ is a stopped-gradient target constructed from target representations and a next action sampled from the latent-conditioned actor. Suppressing ensemble notation, the FB minibatch objective is

$$\mathcal{L}_{\text{FB}} = \frac{\sum_{i \neq j} R_{ij}^2}{2n(n-1)} - \frac{1}{n} \sum_i R_{ii}. \quad (23)$$

The shared backward map is regularized through

$$\mathcal{L}_{\text{orth}} = \frac{\sum_{i \neq j} (b_i^{\top} b_j)^2}{2n(n-1)} - \frac{1}{n} \sum_i \|b_i\|_2^2, \quad b_i = B(s_i^+), \quad (24)$$

and the representation update minimizes $\mathcal{L}_{\text{FB}} + \lambda_{\text{orth}} \mathcal{L}_{\text{orth}}$. Motion-discriminator and auxiliary-critic objectives follow the underlying BFM training procedure; the multi-horizon modification applies to the FB representation and the shared actor objective below.

C. Adaptive Actor Objective

Values at different discounts have different natural scales. We therefore use the normalized horizon-specific objective

$$V_{\gamma}(s, z) = (1 - \gamma) Q_{\gamma}^{\text{FB}}(s, \pi_{\text{BFM}}(s, z), z). \quad (25)$$

The factor $(1 - \gamma)$ removes the leading discounted-sum scale, making the short- and long-horizon terms directly comparable. The shared actor maximizes

$$J_{\text{FB}}(s, z) = \tau \log \sum_{\gamma \in \Gamma} p(\gamma) \exp\left(\frac{V_{\gamma}(s, z)}{\tau}\right), \quad (26)$$

where $p(\gamma) \geq 0$, $\sum_{\gamma} p(\gamma) = 1$, and $\tau > 0$. Actor gradient is

$$\nabla_{\theta} J_{\text{FB}} = \sum_{\gamma \in \Gamma} w_{\gamma}(s, z) \nabla_{\theta} V_{\gamma}(s, z), \quad (27)$$

$$w_{\gamma}(s, z) = \frac{p(\gamma) \exp(V_{\gamma}(s, z)/\tau)}{\sum_{\gamma' \in \Gamma} p(\gamma') \exp(V_{\gamma'}(s, z)/\tau)}. \quad (28)$$

Thus the update remains a convex combination of the horizon-specific policy gradients, but the combination is selected from the current state and latent command rather than fixed globally.

a) Stability of the adaptive FB-to-actor objective:

The relevant stability property is not only that the horizon aggregator is non-expansive, but that it does not amplify

perturbations introduced through the FB backup. Let

$$\tilde{Q}_\gamma = (1 - \gamma)Q_\gamma^{\text{FB}}, \quad (29)$$

and collect the normalized action values as $\tilde{\mathbf{Q}} = [\tilde{Q}_\gamma]_{\gamma \in \Gamma}$. Holding the current backward map B fixed, let r_z^B denote the corresponding latent-conditioned reward. For a fixed latent policy π , the normalized FB Bellman operator is

$$\begin{aligned} (\mathcal{B}^\pi \tilde{\mathbf{Q}})_\gamma(s, a, z) \\ = \mathbb{E} \left[(1 - \gamma)r_z^B(s') + \gamma(1 - d)\tilde{Q}_\gamma(s', \pi(s', z), z) \mid s, a \right]. \end{aligned} \quad (30)$$

Equip the product space with the norm

$$\|\tilde{\mathbf{Q}}\|_{\Gamma, \infty} = \max_{\gamma \in \Gamma} \|\tilde{Q}_\gamma\|_\infty. \quad (31)$$

For any two value collections $\tilde{\mathbf{Q}}$ and $\tilde{\mathbf{Q}}'$, the reward terms cancel, yielding

$$\|\mathcal{B}^\pi \tilde{\mathbf{Q}} - \mathcal{B}^\pi \tilde{\mathbf{Q}}'\|_{\Gamma, \infty} \leq \gamma_{\max} \|\tilde{\mathbf{Q}} - \tilde{\mathbf{Q}}'\|_{\Gamma, \infty}, \quad (32)$$

where $\gamma_{\max} = \max \Gamma < 1$. Thus, for fixed B and π , the multi-horizon FB Bellman backup is a $\gamma_{\max}$ -contraction.

We next define the adaptive actor-value functional

$$(\mathcal{A}_\tau^\pi \tilde{\mathbf{Q}})(s, z) = \tau \log \sum_{\gamma \in \Gamma} p(\gamma) \exp \left(\frac{\tilde{Q}_\gamma(s, \pi(s, z), z)}{\tau} \right). \quad (33)$$

The log-sum-exp map is monotone and translation equivariant. Let

$$\delta = \|\tilde{\mathbf{Q}} - \tilde{\mathbf{Q}}'\|_{\Gamma, \infty}. \quad (34)$$

Then, for every (s, a, z) and $\gamma \in \Gamma$,

$$\tilde{Q}_\gamma(s, a, z) - \delta \leq \tilde{Q}_\gamma(s, a, z) \leq \tilde{Q}_\gamma(s, a, z) + \delta. \quad (35)$$

Monotonicity and translation equivariance therefore imply

$$\|\mathcal{A}_\tau^\pi \tilde{\mathbf{Q}} - \mathcal{A}_\tau^\pi \tilde{\mathbf{Q}}'\|_\infty \leq \|\tilde{\mathbf{Q}} - \tilde{\mathbf{Q}}'\|_{\Gamma, \infty}. \quad (36)$$

Hence the adaptive actor-value functional is non-expansive with respect to perturbations of the horizon-specific FB values.

The same bound holds for the exact improved-value functional

$$(\mathcal{I}_\tau \tilde{\mathbf{Q}})(s, z) = \max_a \tau \log \sum_{\gamma \in \Gamma} p(\gamma) \exp \left(\frac{\tilde{Q}_\gamma(s, a, z)}{\tau} \right), \quad (37)$$

since

$$\left| \max_a f(a) - \max_a g(a) \right| \leq \max_a |f(a) - g(a)|. \quad (38)$$

This statement concerns the improved value functional rather than the arg max policy map itself.

Combining (32) and (36) gives

$$\|\mathcal{A}_\tau^\pi \mathcal{B}^\pi \tilde{\mathbf{Q}} - \mathcal{A}_\tau^\pi \mathcal{B}^\pi \tilde{\mathbf{Q}}'\|_\infty \leq \gamma_{\max} \|\tilde{\mathbf{Q}} - \tilde{\mathbf{Q}}'\|_{\Gamma, \infty}. \quad (39)$$

Therefore, adaptive horizon aggregation preserves the contractive sensitivity of the fixed-policy FB backup: perturba-

TABLE IV: BFM architecture used in our implementation.

Component	Hidden width	Hidden layers
Body actor	2048	6
Forward representation	2048	6
Critics	2048	6
Backward encoder	1024	1
Motion discriminator	1024	3
Executable latent dimension		256
Body action dimension		29
Forward/critic ensemble size		2

tions in the horizon-specific FB values are not amplified by the joint actor objective. Consequently, relative to a single-horizon objective with $\gamma = \gamma_{\max}$, adaptive aggregation introduces no worse worst-case sensitivity to critic perturbations while allowing the shared policy to exploit value estimates at multiple temporal horizons.

b) Adaptive aggregation versus a fixed weighted sum:

A natural multi- γ alternative is the fixed convex average

$$J_{\text{WS}}(s, z) = \sum_{\gamma \in \Gamma} p(\gamma) V_\gamma(s, z). \quad (40)$$

Both J_{WS} and (26) are scale controlled after the $(1 - \gamma)$ normalization; non-expansiveness alone therefore does not distinguish the two. The key difference is that the weighted-sum baseline uses the same horizon weights for every state and behavior, whereas (26) adapts the effective weights to the current horizon-specific values. As $\tau \rightarrow 0$, the update concentrates on the largest normalized value; for large τ , its gradient approaches the prior-weighted average. To isolate this effect from simply exposing the actor to multiple discounts, we additionally compare against (40) using the same multi-horizon representation. The adaptive objective empirically outperforms this fixed weighted-sum baseline, supporting state- and behavior-dependent horizon weighting rather than a fixed global mixture.

For intuition, a discount γ at control interval Δt has effective duration approximately $\Delta t/(1 - \gamma)$. At 50 Hz, $\gamma = 0.8$ corresponds to approximately 0.1 s and $\gamma = 0.98$ to approximately 1 s. This is an interpretation of the temporal weighting, not a hard execution horizon for the latent command.

D. Architecture and Optimization

The actor, forward representations, and critics use residual MLP backbones. The forward representations and critics use two ensemble members; the actor is a single shared network conditioned on the current latent and outputs 29 body actions. The backward encoder maps proprioceptive and privileged kinematic features to the normalized behavioral space. The scalar discount is embedded by a small MLP before entering the forward representation and is not an input to the actor. Details are shown in Tab. IV.

The BFM is frozen during expert projection, projected-data collection, and offline planner training. These stages therefore use the same latent geometry, state convention, and

TABLE V: BFM optimization parameters. Target networks use Polyak updates $\bar{\theta} \leftarrow (1 - \rho)\bar{\theta} + \rho\theta$.

Parameter	Value
Forward-map learning rate	3×10^{-4}
Backward-map learning rate	10^{-5}
Actor / critic learning rate	3×10^{-4}
Discriminator learning rate	10^{-5}
Minibatch size	1024
Training updates	100,000
Horizon set Γ	$\{0.8, 0.98\}$
Horizon prior $p(0.8)/p(0.98)$	$0.625/0.375$
Aggregation temperature τ	1
Replay capacity	5.12×10^6
FB target update rate ρ	0.01
Critic target update rate ρ	0.005
Backward orthogonality coefficient	100

body-action decoder.

APPENDIX II HETEROGENEOUS SKILL ACQUISITION

As in the main paper, skill acquisition is deliberately decoupled from the shared representation. Experts may use different supervision, observations, and training procedures; projection only requires access to the resulting robot actions. We use motion-based and reward-shaped experts for the base interactions, and SF-guided behavior reuse when we specifically want to create diverse compositions between an interaction and native BFM behaviors.

A. Task-Specific Interaction Experts

Motion-based experts combine humanoid motion tracking with object and contact objectives. For box pickup and object grasping, the tracked quantities include both body motion and manipulated-object pose, while contact rewards encourage the demonstrated interaction to be realized physically. The door expert is trained with stage-specific objectives [2] for approach, contact establishment, handle manipulation, and opening. These experts may use different privileged observations and rewards; none of those task-specific inputs are required by the unified planner after projection.

For reproducibility, the HDMI-style motion expert uses a privileged observation encoder with a 256-dimensional embedding and a reference-action residual actor. Its actor hidden widths are [512, 256, 256], critic hidden widths are [512, 256, 128], and the activation is Mish. The PPO settings for this route are summarized in [Tab. VI](#).

B. SF-Guided Behavior Reuse for Skill Composition

The third acquisition route targets interaction–behavior composition. Given an acquired interaction, we use behaviors already represented by the BFM to create additional physically valid variants: the interaction reward preserves the task-specific physical requirements, while a native BFM latent specifies the whole-body behavior to be expressed simultaneously. This expands one interaction into a family of compositions without requiring a separate motion reference or task-specific expert for every combination.

TABLE VI: PPO parameters for the motion expert and the SF-guided composition adapter.

Parameter	Motion expert	SF adapter
Rollout steps / environment	32	32
Learning rate	3×10^{-4}	10^{-5}
Optimization epochs	3	5
Minibatches / rollout	8	4
PPO clipping parameter	0.2	0.2
Discount	0.99	0.99
GAE parameter	0.95	0.95
Entropy coefficient	0.001	0
Gradient-norm limit	1	1

In our experiments, we instantiate this idea with box carrying. By varying the requested native latent, the same carrying interaction can be composed with crouching, kneeling, walking, or running. For example, carrying while crouching requires the controller to maintain support of the box while realizing a substantially different whole-body configuration. The resulting rollouts add diverse compositional behavior to the downstream planner data.

For an interaction I , let $z^N \in \mathcal{Z}$ denote the native BFM behavior to be composed with it. We freeze the BFM and learn an adapter A_{ψ_I} that outputs an executable latent,

$$\hat{z}_t = \Pi_{\mathcal{Z}}(z^N + A_{\psi_I}^{\text{res}}(o_t, z^N)), \quad a_t = \pi_{\text{BFM}}(s_t, \hat{z}_t). \quad (41)$$

The residual parameterization lets the adapter modify the native command only as needed to satisfy the interaction. The deployable adapter uses causal robot inputs and the requested native latent; privileged object state and interaction errors may be used by the critic and reward computation during simulation training.

We optimize the adapter with

$$r_t = r_t^I + \lambda_{\text{SF}} B(s_t)^\top z^N, \quad (42)$$

where r_t^I preserves the interaction and the SF term encourages the rollout to remain aligned with the requested native behavior. For fixed z^N ,

$$\mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t B(s_t)^\top z^N \right] = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t B(s_t) \right]^\top z^N, \quad (43)$$

so the preservation term evaluates the discounted state occupancy along the behavioral direction selected by z^N . The two terms therefore play distinct roles: r_t^I keeps the interaction valid, while the SF term determines which native behavior is composed with that interaction.

For the carrying compositions used in our experiments, r_t^I contains upper-body pose, object-position, object-orientation, and contact terms. The three tracking terms are

$$r_{\text{pose}} = \exp(-e_q/0.25^2), \quad (44)$$

$$r_{\text{pos}} = \exp(-\|e_p\|_2^2/0.10^2), \quad (45)$$

$$r_{\text{ori}} = \exp(-\theta_{\text{obj}}^2/0.35^2), \quad (46)$$

with weights 1, 2, and 1, respectively; the contact term has weight 1. Here e_q is the mean squared upper-body joint-

position error, e_p is the object-position error in the torso frame, and θ_{obj} is the object-orientation discrepancy in that frame. The SF term has weight 0.1, the latent-smoothness penalty has weight -0.5 , and object-drop termination contributes -100 . These interaction terms are shared across the composed variants; behavioral diversity is introduced by changing z^N .

Because both $B(s_t)$ and z^N lie in the radius $\sqrt{d_z}$ sphere,

$$\frac{B(s_t)^\top z^N}{d_z} = \cos \angle(B(s_t), z^N). \quad (47)$$

The normalized and unnormalized forms therefore differ only by a constant scale absorbed into λ_{SF} . Importantly, the preservation target is the requested native behavior z^N , not the adapted latent $\hat{z}_t$.

The adapter actor and critic use two residual feed-forward blocks of width 512 with feed-forward width 2048, GELU activations, and observation normalization. We additionally regularize rapid latent changes with the normalized cost

$$c_{\text{smooth},t} = \frac{\|\hat{z}_t - \hat{z}_{t-1}\|_2^2}{2d_z}, \quad (48)$$

which equals one minus cosine similarity on $\mathcal{Z}$.

APPENDIX III

ACTION-ALIGNED LATENT POLICY PROJECTION

After each expert is acquired, LPP maps its local closed-loop responses into the executable BFM space. This section follows the same order as the main method: first establish the shared action convention, then define the projector and short-horizon latent execution, the action/semantic objectives, and finally the on-policy aggregation used to obtain robust projected rollouts.

A. Common Action Space

Experts may use different normalized action conventions, so action alignment is performed only after both expert and BFM outputs are converted to the same physical body joint-target space. The frozen BFM actor produces $a_t = \pi_{\text{BFM}}(s_t, z_t) \in \mathbb{R}^{29}$. Under the BFM control profile,

$$q_t^{\text{des}} = q^{\text{default}} + 0.25 r_q \odot \text{clip}(6a_t, -6, 6), \quad (49)$$

where r_q contains the per-joint action scales. Joint order, default pose, and scale vectors are kept consistent across expert conversion, projection, projected-rollout collection, and execution. The resulting targets are realized by the robot PD controller with the gains and actuator limits of the selected control profile.

B. Projector and Latent Execution

Each expert is projected independently. At update time t_k , a projector P_{ψ_i} maps the expert observation to BFM latent,

$$z_k = P_{\psi_i}(o_{t_k}^{E_i}), \quad z_k \in \mathcal{Z}. \quad (50)$$

The projector is a pre-normalized residual MLP consisting of an input LayerNorm and linear projection, residual feed-forward blocks, an output LayerNorm, and a latent head

followed by $\Pi_{\mathcal{Z}}$. A residual block is

$$\tilde{h}^\ell = \text{GELU}(W_1^\ell \text{LN}(h^\ell) + b_1^\ell), \quad (51)$$

$$h^{\ell+1} = h^\ell + W_2^\ell \tilde{h}^\ell + b_2^\ell. \quad (52)$$

The hidden width is 1024 with feed-forward expansion ratio four. The motion-based projector uses four residual blocks and the door projector uses two.

Projection updates occur every five body-control steps, corresponding to a 0.1 s interval and a 10 Hz projector. Between two projector updates, the latent command is interpolated on the sphere,

$$z_{t_k+h} = \text{SLERP}(z_{k-1}, z_k, \alpha_h), \quad h = 0, \dots, H_P - 1, \quad (53)$$

where $H_P = 5$. For endpoints $a, b \in \mathcal{Z}$ with angular separation ϑ ,

$$\text{SLERP}(a, b, \alpha) = \frac{\sin((1-\alpha)\vartheta)}{\sin \vartheta} a + \frac{\sin(\alpha\vartheta)}{\sin \vartheta} b. \quad (54)$$

The cosine used to compute ϑ is clamped for numerical stability.

C. Action Alignment and Semantic Regularization

Let T_B and T_i convert the BFM and expert outputs to the same physical body joint-target space. Over the short execution interval beginning at t_k , we use

$$e_h^a = T_B(\pi_{\text{BFM}}(s_{t_k+h}, z_{t_k+h})) - T_i(\pi_i^E(o_{t_k+h}^{E_i})), \quad (55)$$

$$\mathcal{L}_{\text{act}} = \mathbb{E}_{P_{\psi_i}} \left[\frac{\sum_{h=0}^{H_P-1} m_h \|e_h^a\|_2^2}{29 \sum_{h=0}^{H_P-1} m_h} \right], \quad (56)$$

where m_h masks samples beyond an episode boundary. Comparing physical joint targets prevents normalized-action conventions of different experts from changing the meaning of the alignment loss.

Action matching alone can leave the latent underdetermined. We therefore anchor the predicted latent to the native BFM representation of the same local state,

$$\mathcal{L}_{\text{sem}} = \mathbb{E} [\|z_k - B(s_{t_k})\|_2^2]. \quad (57)$$

For two vectors on $\mathcal{Z}$,

$$\|z - B(s)\|_2^2 = 2d_z \left(1 - \frac{z^\top B(s)}{d_z} \right), \quad (58)$$

so squared distance and cosine gap are equivalent up to a constant factor. The full objective is

$$\mathcal{L}_{\text{proj}} = \mathcal{L}_{\text{act}} + \lambda_{\text{sem}} \mathcal{L}_{\text{sem}}. \quad (59)$$

The action term identifies commands by the expert's required local control response, whereas the semantic term resolves action-equivalent solutions in a way that remains coherent with the native BFM geometry. Training details in [Tab. VII](#).

For skills with dexterous hands, a separate projector head regresses the expert's physical hand-joint targets. Its loss is evaluated only when valid hand labels are available. The sphere constraint applies only to the body latent; hand outputs retain their physical joint-target parameterization.

TABLE VII: Latent-projector optimization parameters.

Parameter	Value
Optimizer	AdamW
Learning rate	3×10^{-4}
Weight decay	10^{-4}
Gradient-norm limit	1
Minibatches / rollout	4
Optimization passes / rollout	5
Semantic-loss weight λ_{sem}	0.1
Body-control steps / latent interval H_P	5
Executed latent noise std.	0.1

D. On-Policy Projection and Data Aggregation

Projection is trained with single-skill on-policy data aggregation. The current projector is executed through the frozen BFM under randomized task initializations. At the states it visits, we query the corresponding expert for body actions and record the BFM inputs, expert observations, and reset masks. Short execution windows are then replayed through the frozen BFM actor to evaluate (56) and (57). The projector is updated and new rollouts are collected with the updated policy. Gradients pass through the frozen BFM actor to the predicted latent, but no BFM parameter is changed and no simulator gradient is required. This DAgger-style procedure [35] exposes the projector to corrective states that are absent from nominal expert trajectories.

Exploration noise is applied to the executed latent before projection back to $\mathcal{Z}$ and is never added to the expert action label. Once projection is complete, the original expert is no longer required for projected execution or planner-data collection.

The projected latent should be interpreted locally: it is identified by the control response induced by the frozen BFM on states visited during projection, rather than by assuming that the expert and BFM policies are globally identical. This is why projected closed-loop rollouts, rather than nominal expert trajectories alone, are used as the supervision for the unified planner.

APPENDIX IV UNIFIED LATENT FLOW PLANNING

Once each expert has been projected, its closed-loop latent rollouts can be combined because they now share the same executable representation. We train the unified planner entirely offline on these projected trajectories, matching the data construction, conditioning variables, and flow objective. Details of the planner network are shown in Tab. VIII.

A. Projected Rollout Dataset and Prediction Target

For each projected skill, we collect 20K closed-loop episodes and aggregate them for offline planner training. Each record contains onboard depth, proprioception, the high-level condition, and the action-aligned latent actually executed by the projected policy. Replacing these targets with $B(s)$ would change the supervision from projected control responses to behavioral-state encoding and is therefore not equivalent to our data construction.

The planner predicts a one-second sequence of future BFM commands. We use one latent target every 0.1 s, corresponding to five 50 Hz BFM control steps, and predict $H = 10$ latent commands per sequence. Matching the trajectory notation used in the main paper,

$$\mathcal{Z}_t = [z_t, z_{t+1}, \dots, z_{t+H-1}] \in \mathbb{R}^{H \times d_z}, \quad H = 10, \quad (60)$$

Each future latent is represented by one temporal token in the transformer. For hand-bearing data, a separate target $Y_t \in \mathbb{R}^{H \times 14}$ stores physical hand-joint targets in a fixed hand-major ordering. Hand targets are normalized with their own statistics and are not projected onto $\mathcal{Z}$.

B. Observation and Steering Conditioning

Unlike the task-specific experts, the unified planner uses only onboard observations. Its proprioceptive input is $p_t = [q_t; \dot{q}_t] \in \mathbb{R}^{58}$; for hand-bearing samples, the measured 14 hand-joint positions are included through a separate projection. Continuous features are standardized coordinate-wise using fixed training-set statistics, and the normalization statistics and joint ordering are stored with the model.

a) Depth: The planner processes a metric egocentric depth image of size 144×256 , clipped to the valid range 0.12–3.0 m. A strided convolution with 12×16 kernel and stride produces a 12×16 grid of 192 spatial tokens. Learned spatial embeddings are added before two self-attention blocks. Eight learned register tokens participate in the depth encoder and are discarded afterward, leaving the 192 spatial tokens as conditioning memory.

b) Proprioception: The 58-dimensional joint state is standardized coordinate-wise and linearly embedded to the model width. For coordinate j , $\bar{p}_j = (p_j - \mu_j) / \max(\sigma_j, \epsilon)$. When hands are enabled, their measured joint positions are standardized separately and added through an additional projection.

c) Language: Task instructions are encoded by a frozen T5-base text encoder. Masked pooling produces a 768-dimensional sentence embedding, followed by LayerNorm and a two-layer SiLU MLP that maps it to the planner width. The text encoder is not updated by the planner objective.

d) Waypoint: Waypoint supervision is derived from future root poses. At horizons $\delta \in \{0.5, 1.0\}$ s, we use

$$u_t^\delta = [\Delta x_t^\delta, \Delta y_t^\delta, \sin \Delta \psi_t^\delta, \cos \Delta \psi_t^\delta, h_{t+\delta}], \quad (61)$$

where displacement is expressed in the current heading frame, $\Delta \psi$ is relative yaw, and h is future pelvis height. Concatenating the two horizons gives a ten-dimensional waypoint condition. Missing waypoint conditions are masked rather than represented by a zero-valued physical command.

e) Steering representation: Language and waypoint embeddings are converted to a small fixed set of steering tokens by a self-attention encoder. The latent transformer cross-attends to these steering tokens, the 192 spatial depth tokens, one proprioceptive state token, and eight learned conditioning-register tokens. Unavailable language or waypoint modalities are masked explicitly.

TABLE VIII: Architecture of the unified latent flow planner.

Component or parameter	Configuration	Component or parameter	Configuration
Executable body latent	$d_z = 256$	Transformer width / latent blocks	1024 / 16
Prediction horizon	1 s	Attention heads / per-head width	8 / 128
Body / optional hand state	58 / 14	Feed-forward expansion / activation	4 / GELU
Optional hand target	14 per latent step	Attention / feed-forward dropout	0 / 0.2
Depth image / patch shape	$144 \times 256 / 12 \times 16$	Conditioning registers	8
Depth range / spatial tokens	0.12–3.0 m / 192	Flow-time embedding scale	1000
Depth-encoder blocks / registers	2 / 8	Pooled language embedding	768 (frozen T5-base)
Waypoint input	10, at 0.5 and 1.0 s	Steering blocks / output tokens	2 / 8

C. Conditional Latent Transformer

Each noisy future latent is linearly embedded and combined with a learned temporal-position embedding. If hand targets are present, a separate projection of the noisy hand target is added to the corresponding temporal token. Flow time is sinusoidally embedded and passed through an MLP. Each transformer block contains temporal self-attention, cross-attention to the observation/steering memory, and a feed-forward MLP. Adaptive LayerNorm modulation supplies shift, scale, and residual-gate vectors from the flow-time condition. For example,

$$h' = h + g_\tau \odot \text{Attn}((1 + s_\tau) \odot \text{LN}(h) + b_\tau). \quad (62)$$

Separate linear heads decode the clean body latents and optional hand targets.

D. Offline Flow-Matching Objective

Let ξ_t denote the encoded onboard observations and available steering conditions. We sample $x_0 \sim \mathcal{N}(0, I)$ and $\tau \sim \mathcal{U}(0, 1)$, and construct

$$x_\tau = (1 - \tau)x_0 + \tau x_1. \quad (63)$$

The network predicts the clean endpoint $f_\eta(x_\tau, \tau, \xi_t)$ and is trained with

$$\mathcal{L}_{\text{body}} = \mathbb{E} \left[\frac{1}{Hd_z} \|f_\eta(x_\tau, \tau, \xi_t) - x_1\|_2^2 \right]. \quad (64)$$

The clean latent targets lie on $\mathcal{Z}$, but the Gaussian interpolation and raw endpoint prediction are carried out in Euclidean coordinates. Sphere projection is applied only when sampled outputs are converted to executable BFM commands.

An analogous Gaussian interpolation is applied to the hand targets when available. The complete loss is

$$\mathcal{L}_{\text{planner}} = \mathcal{L}_{\text{body}} + \lambda_{\text{hand}} \mathcal{L}_{\text{hand}}, \quad (65)$$

where $\mathcal{L}_{\text{hand}}$ is evaluated only on valid hand samples. No decoded body-action auxiliary loss is used in the reported planner training; action alignment is already incorporated when constructing the projected latent supervision.

An endpoint predictor induces the velocity field

$$v_\eta(x_\tau, \tau, \xi_t) = \frac{f_\eta(x_\tau, \tau, \xi_t) - x_\tau}{1 - \tau}, \quad \tau < 1. \quad (66)$$

For a numerical grid $0 = \tau_0 < \dots < \tau_N = 1$, Euler

TABLE IX: Offline planner optimization parameters.

Parameter	Value
Optimizer	AdamW
Learning rate	2×10^{-4}
Weight decay	10^{-2}
Warm-up updates	100
Post-warm-up schedule	Constant
Batch size	256
Training epochs	500
Gradient-norm limit	1
Body endpoint-loss weight	1
Hand endpoint-loss weight	1
Decoded body-action loss weight	0
Default windows / worker	16
Validation fraction	0.2
Flow-time distribution	$\mathcal{U}(0, 1)$
Noise distribution	$\mathcal{N}(0, I)$

integration uses

$$x_{j+1} = x_j + (\tau_{j+1} - \tau_j)v_\eta(x_j, \tau_j, \xi_t). \quad (67)$$

Heun integration uses the corresponding predictor-corrector update. After sampling, every body latent is independently projected by $\Pi_{\mathcal{Z}}$, while hand outputs are denormalized using their own statistics.

E. Planner Optimization

Shown in [Tab. IX](#), training windows are sampled within episode boundaries, and train/validation partitions are defined at the episode level. The model checkpoint stores the architecture, optimizer state, proprioceptive normalization, and hand-target normalization. In the exported configuration, state-condition dropout is 0.2 and waypoint-condition dropout is 0.1; waypoint tokens are omitted for episodes without waypoint supervision.

F. Receding-Horizon Execution

At deployment, the planner executes the leading segment of the sampled latent sequence and then replans from the new onboard observation. Consecutive latent updates are connected by SLERP to supply the BFM with a command at each 50 Hz body-control step. The BFM actor therefore retains high-frequency state-dependent feedback while the planner operates at a lower rate. Hand joint targets follow the same receding-horizon schedule as the corresponding body-latent sequence.

TABLE X: Simulation evaluation variations.

Task	Variation	Evaluation range
Box pickup	Box geometry, placement, and mass	Base size $20 \times 30 \times 40$ cm, independently scaled by $s_x, s_y \in [0.5, 1.5]$ and $s_z \in [0.5, 1.0]$, yielding sizes from $10 \times 15 \times 20$ cm to $30 \times 45 \times 40$ cm. Boxes are placed either on the ground or on supports 0.25–0.50 m high, with mass sampled from 0.5–1.5 kg.
Door opening	Door geometry and handle placement	Door width 0.8–1.1 m and height 1.9–2.2 m. Handle height is 0.85–0.95 m above the ground, with lateral inset 0.08–0.15 m.
Object grasping	Object identity and shape	128 distinct object instances are evaluated with four takes per object (512 episodes total), selected through teacher-feasibility screening from 685 object instances and 2,991 usable demonstrations.

APPENDIX V

EVALUATION AND REAL-WORLD DEPLOYMENT

A. Simulation Evaluation Protocol

We evaluate each task over diverse object or environment variations in IsaacSim. For pickup and door opening, the *w/ appr.* setting initializes the humanoid approximately 0.5–1.5 m from the interaction target and requires it to approach before execution, whereas *w/o appr.* initializes the humanoid directly in front of the target. Pickup succeeds if the box is lifted by at least 0.3 m and maintained for at least 1 s; grasping succeeds if the object is lifted by at least 0.1 m and maintained for at least 1 s. Door opening succeeds when the door is opened and the humanoid passes through the doorway. Reported $\pm$ values are computed from repeated evaluations with different random seeds to account for simulation stochasticity. Detailed object variation configurations used in evaluation are shown in [Tab. X](#).

B. Real-World Execution

The real-world system uses onboard RealSense depth and proprioception. Both the unified planner and the frozen BFM run on a single NVIDIA RTX PRO 5000 Blackwell laptop. The TensorRT planner replans at 5 Hz using 10 Heun integration steps, with approximately 20 ms inference time per planner call, while the BFM runs at 50 Hz with approximately 1 ms per forward pass.

We use RTC during deployment to avoid discontinuities between consecutively sampled latent trajectories. At each replanning step, the flow planner predicts a chunk of future latent commands while the robot continues executing the current chunk. The newly generated chunk overlaps temporally with the remaining portion of the previous prediction, and RTC conditions the overlapping prefix of the new sample toward the previously committed trajectory during flow integration. This preserves continuity across replanning boundaries while still allowing later latent commands in the chunk to adapt to the latest onboard observation. In this way, RTC decouples the planner inference latency from continuous robot execution and avoids abrupt latent changes caused by independently sampling each plan.