Learning Versatile Humanoid Manipulation with Touch Dreaming

Yaru Niu[1,3], Zhenlong Fang[1], Binghong Chen[1], Shuai Zhou[1], Revanth Krishna Senthilkumaran[1,3], Hao Zhang[1,2], Bingqing Chen[3], Chen Qiu[3], H. Eric Tseng[2], Jonathan Francis[1,3], and Ding Zhao[1]
[1]Carnegie Mellon University, [2]UT Arlington, [3]Bosch Center for AI
humanoid-touch-dream.github.io

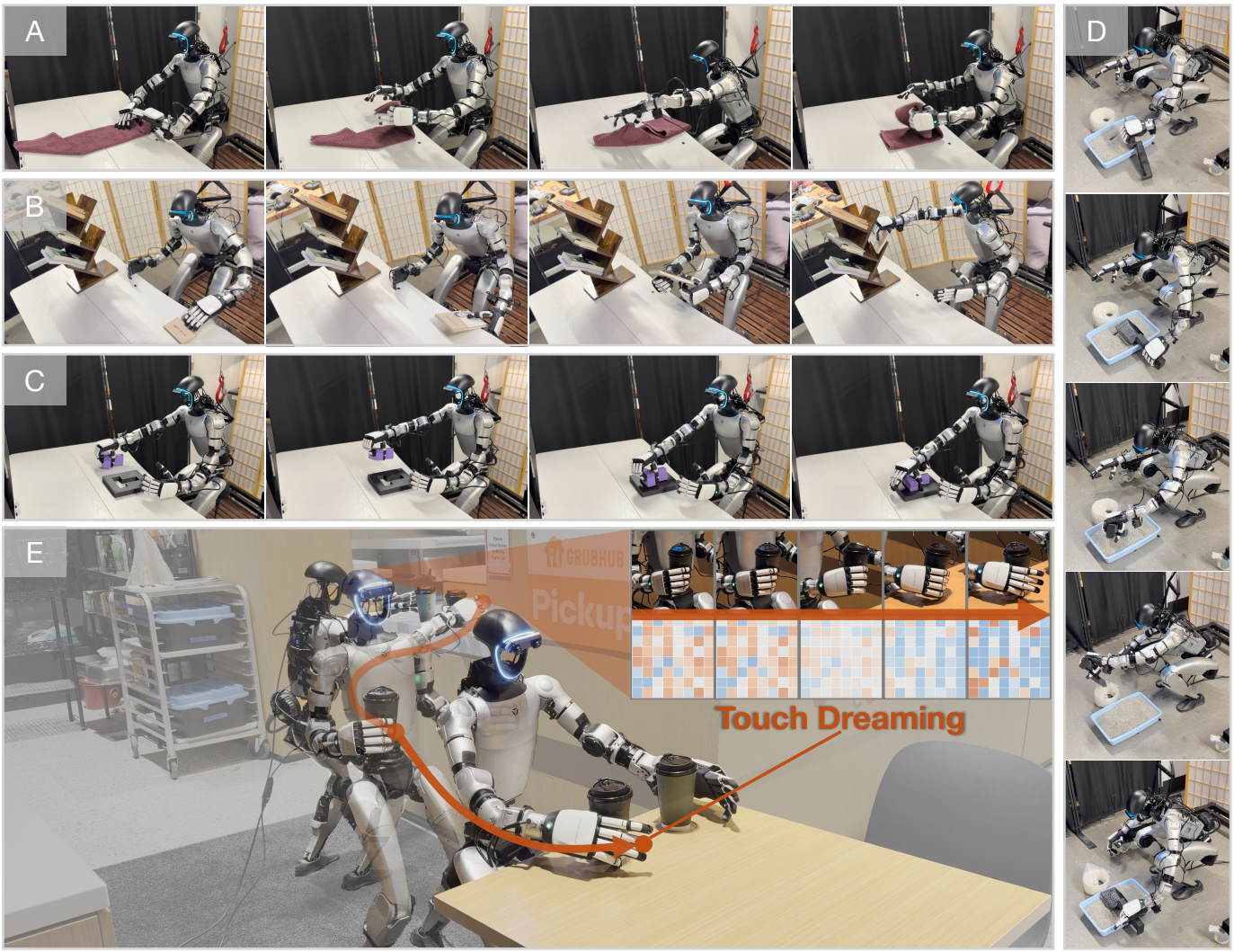


Fig. 1: Our system enables **versatile**, **contact-rich**, and **dexterous** humanoid manipulation. *A:* long-horizon, multi-stage manipulation of deformable objects (towel folding). *B:* mixed prehensile and non-prehensile manipulation for thin-profile rigid objects with limited grasp affordance (book organization). *C:* tight-tolerance insertion with a clearance of 3.5 mm, requiring high precision and reactive adaptation (Insert-T). *D:* dexterous, tool-mediated contact under low-profile constraints (cat litter scooping). *E:* bimanual object fetch and loco-manipulation, requiring stable whole-body transport while keeping objects balanced (tea serving); the embedded panel visualizes **touch dreaming** by showing the dreamed tactile latent of the right middle finger as a normalized heatmap that changes with finger-object contact.

***Abstract*—Humanoid robots promise general-purpose assistance, yet real-world humanoid loco-manipulation remains challenging because it requires *whole-body stability*, *end-effector dexterity*, and *contact-aware interaction* under frequent contact changes. In this work, we study dexterous, contact-rich humanoid loco-manipulation. We first develop an RL-based lower-body controller that serves as the stability backbone for whole-body execution during complex manipulation. Built on this controller, we develop a VR-based whole-body humanoid data collection system that integrates dexterous hands and tactile sensing for contact-rich manipulation. We then propose Humanoid Transformer with Touch Dreaming (HTD), a multimodal encoder–decoder Transformer that models touch as a core modality alongside multi-view vision and proprioception. HTD is trained in a single**

stage with behavioral cloning augmented by *touch dreaming*: in addition to predicting action chunks, the policy predicts future hand-joint forces and future tactile latents, with tactile-latent targets provided by an exponential moving average target encoder without requiring a separate tactile pretraining stage. This encourages the policy to learn contact-aware representations for dexterous manipulation. Across five real-world contact-rich tasks, HTD achieves a 90.9% relative improvement in average success rate over the stronger baseline. Ablation results further show that latent-space tactile prediction is more effective than raw tactile prediction, yielding a 30% relative gain in success rate. These results demonstrate that our touch-dreaming-enhanced learning system enables versatile, high-dexterity humanoid manipulation in the real world. Project page: humanoid-touch-dream.github.io.

I. Introduction

Humanoid robots promise general-purpose physical assistance, fueled by rapid progress in whole-body control, teleoperation, and humanoid learning systems [14, 29, 44, 46, 11, 32]. Yet real-world humanoid loco-manipulation remains fundamentally challenging because it requires tight coordination between whole-body stability and end-effector dexterity under frequent contact changes. Unlike fixed-base platforms, humanoid manipulation is physically coupled with torso posture, base motion, and foot-ground stability, so uncertain hand-object contact can affect not only local manipulation accuracy but also whole-body execution. Consequently, accurate hand motion alone is insufficient; reliable humanoid manipulation requires stable whole-body coordination together with contact-aware interaction.

Despite these requirements, few existing systems provide a practical real-world pipeline that jointly supports stable whole-body execution, full dexterous-hand control, and tactile sensing. Although recent humanoid systems have improved motion tracking, teleoperation, and demonstration collection [52, 28, 59], Table I highlights that few systems combine *whole-body control*, *full end-effector dexterity*, and *touch sensing/modeling* in a single platform for dexterous, contact-rich manipulation. To address this, we build an integrated **whole-body humanoid manipulation system** that combines VR teleoperation with an RL-based lower-body controller (LBC), upper-body IK, dexterous hand retargeting, distributed tactile sensing, and touch modeling. This design provides a stable execution backbone for collecting high-quality real-world demonstrations while allowing the operator to focus on task intent and dexterous interaction.

Built on this tactile-enabled system, the next challenge is how to learn contact-aware policies from multimodal humanoid demonstrations. Purely action-supervised behavioral cloning from vision and proprioception can struggle in contact-rich manipulation, where contact is partially observed and changes abruptly over time [24]. Tactile sensing provides a natural complementary signal for capturing these contact dynamics, and prior work has demonstrated its value in visuo-tactile manipulation and predictive tactile learning [5, 17, 47, 58]. Yet most existing tactile learning methods are developed for arm-hand manipulation and often rely on separate tactile pretraining, explicit world-model modules, multi-stage inference, or manually designed virtual targets tied to specific sensing and control setups [56, 47, 58, 49, 8]. More broadly, predictive latent learning in Joint-Embedding Predictive Architectures such as I-JEPA [1] and V-JEPA2 [2] suggests that future prediction in latent space can induce semantically meaningful representations without reconstructing raw observations or training a separate generative pipeline. However, these ideas have rarely been brought into a single-stage whole-body humanoid imitation policy that must jointly handle dexterous manipulation, locomotion-related action generation, and rapidly changing contact.

Motivated by this gap, we propose **Humanoid Transformer with Touch Dreaming (HTD)**, a multimodal encoder–decoder Transformer for dexterous humanoid loco-manipulation. HTD models touch as a core modality alongside multi-view vision and proprioception, and is trained in a *single stage* with behavioral cloning augmented by **touch dreaming**. In addition to predicting action chunks, HTD predicts future hand-joint forces and future tactile latents. The tactile targets are produced by an Exponential Moving Average (EMA) target encoder, yielding stable latent supervision without requiring a separate tactile pretraining stage. Rather than using future touch prediction as a separate world model or inference-time module, HTD uses it as an auxiliary objective that regularizes the shared Transformer trunk to learn contact-aware latent dynamics while keeping deployment simple.

We evaluate the full system on five real-world contact-rich tasks: *Insert-T*, *Book Organization*, *Towel Folding*, *Cat Litter Scooping*, and *Tea Serving*. These tasks span tight-tolerance insertion, hard-to-grasp rigid-object manipulation, long-horizon deformable-object handling, low-profile tool use, and bimanual loco-manipulation, together stressing precise alignment, sustained contact, whole-body coordination, and diverse interaction modes. Across these tasks, HTD achieves a **90.9%** relative improvement in average success rate over the stronger ACT baseline, while ablations show that latent tactile dreaming is more effective than raw tactile prediction. Together, these results suggest that combining stable whole-body execution, tactile-enabled dexterous manipulation, and predictive touch dreaming provides a practical path toward reliable humanoid manipulation under frequent contact changes.

Our contributions are threefold:

- We develop a tactile-enabled whole-body humanoid manipulation system for stable, dexterous, contact-rich real-world manipulation.
- We introduce Humanoid Transformer with Touch Dreaming (HTD), a multimodal encoder–decoder Transformer for humanoid loco-manipulation that treats touch as a core modality and is trained in a single stage with touch dreaming, including future hand-joint-force prediction and EMA-supervised tactile-latent prediction for contact-aware representation learning.
- We evaluate the full framework on five real-world contact-rich humanoid manipulation tasks and show that HTD consistently improves over baselines, with latent tactile prediction outperforming raw tactile prediction.

II. Related Work

TABLE I: Comparisons to previous humanoid manipulation learning systems

Method	End-Effector Dexterity	Whole-Body	Touch Sensing	Touch Modeling
OmniH2O [14]	Dex-Hand Full	✓	✗	✗
HumanPlus [11]	Dex-Hand Full	✓	✗	✗
Mobile-TeleVision [31]	Dex-Hand Full	✓	✗	✗
AMO [25]	Dex-Hand Full	✓	✗	✗
ViTacFormer [17]	Dex-Hand Full	✗	✓	✓
TWIST2 [52]	Dex-Hand Open/Close	✓	✗	✗
ViTac Humanoid [23]	Dex-Hand Full	✗	✓	✗
SONIC [32]	Dex-Hand Open/Close	✓	✗	✗
Humanoid UMI [35]	Gripper	✓	✗	✗
HumDex [18]	Dex-Hand Full	✓	✗	✗
Ours	Dex-Hand Full	✓	✓	✓

A. Humanoid Whole-Body Control and Teleoperation for Manipulation

Recent progress in humanoid manipulation has been enabled by advances in whole-body control, motion tracking, and teleoperation infrastructure. A central question in humanoid whole-body control is how to represent and execute task commands across diverse behaviors, including locomotion, loco-manipulation, and upper-body manipulation. Prior works instantiate different control interfaces, such as root tracking, joint-space tracking, and body-keypoint or pose tracking, depending on the task and operator interface [15, 11, 10, 14]. One line of work improves robustness through decomposition, separating functions such as lower-body stabilization, upper-body tracking, force adaptation, or compliance modulation, as in dual-agent force-adaptive control [55], heterogeneous meta-control over multiple control modes [43], adaptive compliance control [7], and hybrid optimization-and-learning frameworks for dexterous whole-body behaviors [25, 31]. Related systems also combine learned whole-body control with specialized teleoperation hardware or tracking modules for more precise loco-manipulation [9, 3]. Another line instead seeks unified controllers that directly coordinate locomotion and manipulation within a single whole-body tracking framework [41, 50]. Complementary teleoperation and motion-tracking systems further improve the practicality and scalability of commanding humanoids through RGB- or pose-based shadowing, immersive VR interfaces, portable mocap-free setups, and closed-loop long-horizon tracking [11, 15, 14, 10, 52, 35, 32, 27, 28, 59]. Building on this line of work, our system combines an RL-based whole-body controller with a VR-based teleoperation stack using a unified reference frame, upper-body IK, and hand retargeting, enabling efficient collection of whole-body humanoid manipulation demonstrations for downstream policy learning.

B. Imitation Learning for Humanoid Manipulation

Built on these advances, recent work has made humanoid manipulation increasingly learnable from demonstrations. Systems such as HumanPlus [11] and OmniH2O [14] couple real-world whole-body teleoperation with behavior cloning, while newer approaches improve scalability and generalization through portable data collection, stronger policy parameterizations, and human-data supervision, including TWIST2 [52], Choice Policies [37], 3D diffusion policies [51], robot-free demonstration interfaces [35], single-video imitation [26], human-humanoid co-training [38], and pretrain-then-finetune pipelines for dexterous humanoid manipulation [18]. Together, these works substantially reduce the barrier to learning whole-body humanoid skills beyond small-scale robot-only behavior cloning.

Table I highlights a remaining gap. Prior humanoid systems such as OmniH2O [14], HumanPlus [11], Mobile-TeleVision [31], AMO [25], TWIST2 [52], SONIC [32], and HumDex [18] support whole-body humanoid manipulation with varying levels of end-effector dexterity, while Humanoid UMI [35] focuses on gripper-based whole-body manipulation learning from robot-free data. Yet most do not incorporate tactile sensing, and fewer still explicitly model tactile signals in the learned policy. Conversely, touch-centric works such as ViTacFormer [17] and the humanoid visual-tactile-action dataset of Kwon *et al.* [23] demonstrate the value of tactile information, but do not provide a learned humanoid manipulation system that combines whole-body control, full end-effector dexterity, tactile sensing, and touch modeling. Our method targets this missing intersection by learning a single-stage touch-aware humanoid manipulation policy with full end-effector dexterity, tactile sensing, and implicit touch modeling through future-touch prediction.

C. Representation Learning for Contact-Rich Manipulation with Tactile Sensing

Tactile sensing has increasingly been studied as a representation-learning problem rather than only a task-specific perception module. Early visuo-tactile manipulation works showed that touch complements vision for resolving contact state under partial observability [5, 24]. More recent work learns transferable tactile representations across sensors, tasks, and embodiments, improving data efficiency and reuse in downstream manipulation [19, 56]. In parallel, a growing line of visuo-tactile action models incorporates touch or force directly into policies for contact-rich manipulation, including diffusion-based, transformer-based, and VLA-style approaches [16, 45, 17, 21, 60, 6, 30, 49, 58, 54, 8, 22, 4, 53]. These works consistently suggest that touch provides critical information about force, slip, compliance, and contact transitions that is difficult to infer from vision alone.

A closely related direction uses *predictive* tactile learning to improve contact-aware representations. Prior work has explored self-supervised multimodal prediction for contact-rich tasks [24], while more recent methods explicitly predict future tactile observations, tactile latents, or related contact quantities [17, 47, 20, 58, 49, 54, 8, 48]. These methods show that anticipating future touch can improve representation quality, planning, or reactive control. Some also rely on manually designed virtual targets tied to specific tactile sensor layouts [49, 8]. Our method instead learns directly from future hand forces and EMA-supervised tactile latents,

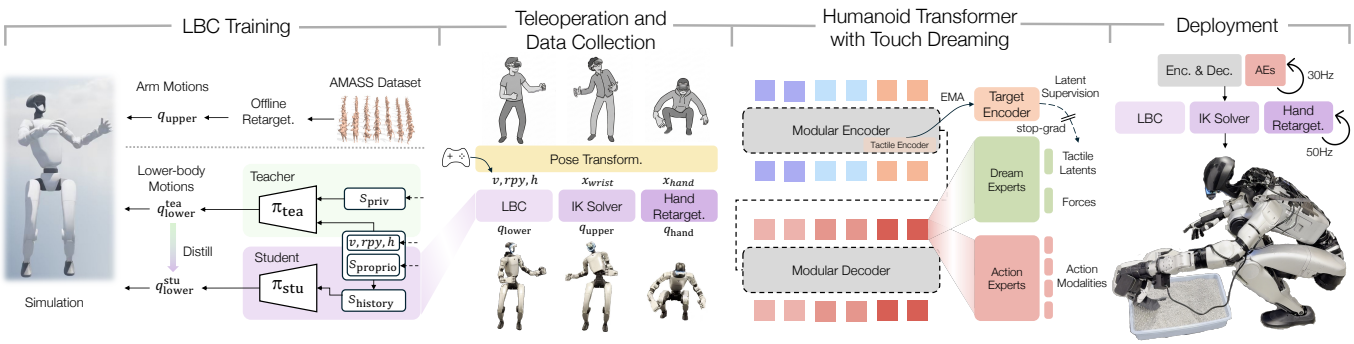


Fig. 2: **System Overview. Left (LBC Training):** A teacher-student framework trains the lower-body controller (LBC) to track base velocity, torso orientation, and height, while robustly handling retargeted arm motions from the AMASS dataset. **Middle-Left (Teleoperation):** Human VR motions are mapped into unified torso commands (for LBC), end-effector poses (for IK), and hand targets (for retargeting), with a joystick dictating base velocity. **Middle-Right (Touch Dreaming):** A multi-modal transformer policy processes vision, touch, and proprioception to predict action chunks alongside future hand joint forces and tactile latents. Future tactile latents are supervised by an EMA target encoder (teacher encoder in Sec. III-E) with stop-gradient, providing stable latent targets. **Right (Deployment):** The policy streams action chunks at 30 Hz to the LBC, IK solver, and hand retargeter, all of which operate at 50 Hz.

avoiding such sensor-specific target engineering. At the same time, much of this literature focuses on arm-hand manipulation and often relies on separate tactile pretraining, explicit world-model modules, or multi-stage inference in which predicted tactile signals are fed into a downstream policy or planner [56, 47, 58].

In contrast, we use future-touch prediction not as a separate world model or inference-time module, but as an auxiliary objective inside a single-stage whole-body humanoid imitation policy. Our framework augments behavioral cloning with *touch dreaming*: prediction of future hand forces together with future tactile latents supervised by an EMA teacher. This regularizes the shared Transformer trunk to learn contact-aware latent dynamics while keeping both training and deployment simple. Unlike prior work centered on arm-hand systems or multi-stage visuo-tactile pipelines, our method integrates future-touch prediction directly into a single-stage policy for dexterous, contact-rich *whole-body humanoid* manipulation.

III. Methodology

A. A System for Versatile Humanoid Dexterous Manipulation

Fig. 2 presents our system for learning real-world, dexterous, contact-rich humanoid manipulation. The system consists of four stages: lower-body controller (LBC) training, VR-based teleoperation and data collection, policy learning with Humanoid Transformer with Touch Dreaming (HTD), and deployment. At its foundation is an RL-based LBC that provides stable lower-body and torso execution during manipulation. We train this controller in simulation with a teacher–student framework: a teacher policy learns robust lower-body behaviors under retargeted arm motions, and a deployable student policy imitates it using only proprioception and short history. The resulting LBC tracks base velocity, torso orientation, and height commands, and serves as the execution backbone during both teleoperation and HTD policy deployment.

Built on this controller, we collect whole-body humanoid demonstrations through VR teleoperation. Human head, wrist, and hand motions are transformed into a unified robot reference frame and decomposed into torso commands for the LBC, end-effector pose targets for an IK solver, and hand targets for dexterous retargeting; the operator additionally provides base velocity commands through a joystick. The resulting dataset contains synchronized camera views, proprioception, hand-force signals, and tactile observations paired with whole-body action targets.

Using these demonstrations, we train HTD, a multimodal touch-aware loco-manipulation policy. HTD uses a modular encoder–decoder Transformer to tokenize multi-view images, robot and hand proprioception, hand-force signals, and tactile inputs into a shared latent representation, and to decode structured action outputs for the body and hands. In addition to action chunk prediction, HTD introduces touch-dreaming heads that predict future hand joint forces and future tactile latents. HTD is trained in a single stage with behavioral cloning augmented by these auxiliary touch-dreaming objectives. Future tactile latents are supervised by an EMA target encoder, which provides stable latent targets without requiring a separate tactile pretraining stage; gradients are stopped through the EMA encoder so it serves only as a slowly evolving target network. These auxiliary objectives regularize the shared Transformer trunk to learn contact-aware latent dynamics. During deployment, the policy streams action chunks to the LBC, IK solver, and hand retargeter, while the dream heads are used only during training and are not executed at inference time.

B. Lower-body Controller

We train the humanoid lower-body policy in massively parallel simulation with IsaacLab [34]. The lower-body policy is command-conditioned and aims to track base motion and torso pose targets. At each control step t, the deployable proprioceptive observation $\boldsymbol{s}^t_{\text{proprio}}$ is defined as

$$\boldsymbol{s}^t_{\text{proprio}} = \left[\boldsymbol{\omega}^t,\ \boldsymbol{g}^t,\ \boldsymbol{q}^t_{\text{lower}},\ \dot{\boldsymbol{q}}^t_{\text{lower}},\ \boldsymbol{a}^{t-1}_{\text{lower}}\right]. \tag{1}$$

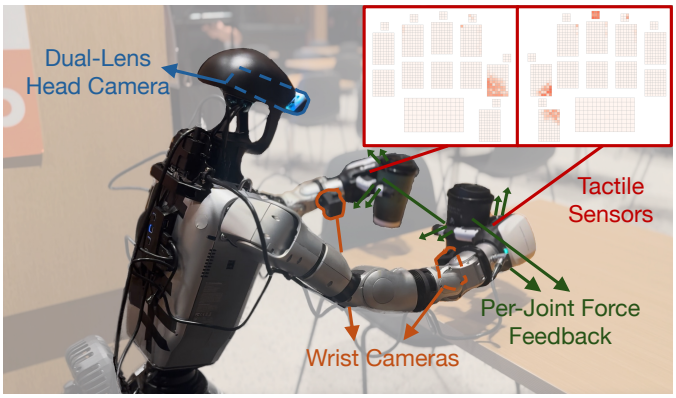


Fig. 3: **System setup.** Hardware used for whole-body humanoid data collection and policy learning, including a dual-lens head camera, wrist cameras, dexterous hands equipped with distributed tactile sensors, and per-joint force feedback from the hand joints. The tactile layout covers the fingers and palm on both hands, and the inset visualizes the corresponding sensor maps together with representative contact activations.

Here, $\boldsymbol{\omega}^t$ is the base angular velocity and $\boldsymbol{g}^t$ denotes the projected gravity vector, both expressed in the body frame; $\boldsymbol{q}^t_{\text{lower}}$ and $\dot{\boldsymbol{q}}^t_{\text{lower}}$ are the lower-body joint positions and velocities, and $\boldsymbol{a}^{t-1}_{\text{lower}}$ is the previous lower-body action. The action output $\boldsymbol{q}_{\text{lower}} \in \mathbb{R}^{15}$ is a 15-dimensional vector of target joint positions: 2×6 for the two legs and 3 for the waist motors.

We adopt a teacher–student framework to train the lower-body policy. The teacher policy is first trained in simulation using PPO [40] with access to privileged information, and is subsequently distilled into a student policy via DAgger [39]. The student policy observes only information available in the real world and can therefore be deployed for both teleoperation and autonomous execution. During training, the upper-body joints are not controlled by this policy; instead, we replay retargeted arm joint references sampled from AMASS [33] to simulate the torques and disturbances induced by upper-body manipulation.

Teacher policy. The teacher policy is formulated as

$$\pi^T(\boldsymbol{s}_{\text{proprio}},\ \boldsymbol{s}_{\text{priv}},\ \boldsymbol{v},\ \boldsymbol{rpy},\ h) = \boldsymbol{q}^T_{\text{lower}}, \tag{2}$$

where $\boldsymbol{q}_{\text{lower}} \in \mathbb{R}^{15}$ represents the 15-DoF lower-body target joint positions. The privileged observation $\boldsymbol{s}_{\text{priv}} = \boldsymbol{c}_{\text{feet}}$ is a binary foot-contact indicator $\boldsymbol{c}_{\text{feet}} \in \{0,1\}^2$ available in simulation. The command inputs include the base velocity $\boldsymbol{v}$, the torso orientation $\boldsymbol{rpy}$, and the base height h. The teacher policy maximizes a weighted sum of tracking rewards for commanded base motion and torso pose, together with regularization, contact, gait, stability, and termination terms.

Student policy. We distill the teacher into a deployable student policy via DAgger. The student policy consumes only real-world-available observations:

$$\pi^S(\boldsymbol{s}_{\text{proprio}},\ \boldsymbol{s}_{\text{history}},\ \boldsymbol{v},\ \boldsymbol{rpy},\ h) = \boldsymbol{q}^S_{\text{lower}}, \tag{3}$$

To compensate for partial observability, the student concatenates a 2-timestep history of proprioceptive observations $\boldsymbol{s}_{\text{history}}$. During training, the student rolls out its own actions in simulation while being supervised by the teacher's reference actions at each timestep, minimizing the L_2 loss $\mathcal{L} = \|\boldsymbol{q}^S_{\text{lower}} - \boldsymbol{q}^T_{\text{lower}}\|_2^2$ between student and teacher outputs.

Training details. During training, command signals are uniformly sampled from predefined ranges to cover diverse loco-manipulation behaviors. We also apply domain randomization to improve sim-to-real transferability. Additional details of LBC training are provided in Sec. V-A in Appendix.

C. Teleoperation and Data Collection

As summarized in Fig. 2, our demonstration pipeline couples VR-based motion mapping with whole-body command execution to collect synchronized humanoid trajectories in real-world settings. At runtime, the operator's head, wrist, and hand motions are transformed from the VR frame into a unified robot reference frame. From these signals, we derive torso pose commands $(\boldsymbol{rpy}, \boldsymbol{h})$, 6D wrist pose targets $\boldsymbol{x}_{\text{wrist}}$ for upper-body execution, and hand targets $\boldsymbol{x}_{\text{hand}}$ for dexterous retargeting. The base velocity command $\boldsymbol{v}$ is provided separately through a joystick. This design lets the operator focus on task intent and dexterous interaction, while the robot-side control stack handles stabilization and low-level execution.

These targets are executed through a three-stage stack. First, the LBC takes $(\boldsymbol{v}, \boldsymbol{rpy}, \boldsymbol{h})$ and produces lower-body joint targets $\boldsymbol{q}_{\text{lower}}$ to maintain stable locomotion, posture, and torso tracking. Second, an IK solver maps the desired wrist/end-effector poses $\boldsymbol{x}_{\text{wrist}}$ to upper-body joint targets $\boldsymbol{q}_{\text{upper}}$. Third, a hand retargeting module based on DexPilot [12] converts the human hand targets $\boldsymbol{x}_{\text{hand}}$ into dexterous hand joint targets $\boldsymbol{q}_{\text{hand}}$ by optimizing fingertip-distance consistency for reliable grasping and in-hand interaction. Together, this stack enables coordinated whole-body teleoperation while preserving full end-effector dexterity.

During teleoperation, we record synchronized multimodal observations from the humanoid, including RGB images from the dual-lens head camera and wrist cameras, robot and hand proprioception, per-joint force feedback from the dexterous hands, and tactile readings from both hands. Each hand provides a 1062-dimensional tactile observation distributed across 17 spatial sensing regions spanning the finger segments and palm surfaces. This distributed tactile layout captures localized contact patterns over the hand surface, as visualized in Fig. 3. The resulting dataset pairs whole-body action commands $(\boldsymbol{v}, \boldsymbol{rpy}, \boldsymbol{h},\ \boldsymbol{x}_{\text{wrist}},$ and $\boldsymbol{x}_{\text{hand}})$ with multi-view vision, robot and hand proprioception, per-joint hand-force feedback, and distributed tactile observations for downstream policy learning.

D. Learning Dexterous Manipulation with Touch Dreaming

We aim to learn a versatile humanoid manipulation policy that robustly handles contact-rich interactions by modeling *touch* as a core modality. We introduce **Humanoid Transformer with Touch Dreaming (HTD)**, shown in Fig. 4. HTD follows a modular design with three groups of components: (i) modality tokenizers that encode each observation stream into tokens, (ii) an encoder–decoder transformer trunk that fuses multimodal information and model complex dynamics, and (iii) modular experts that decode the trunk outputs into

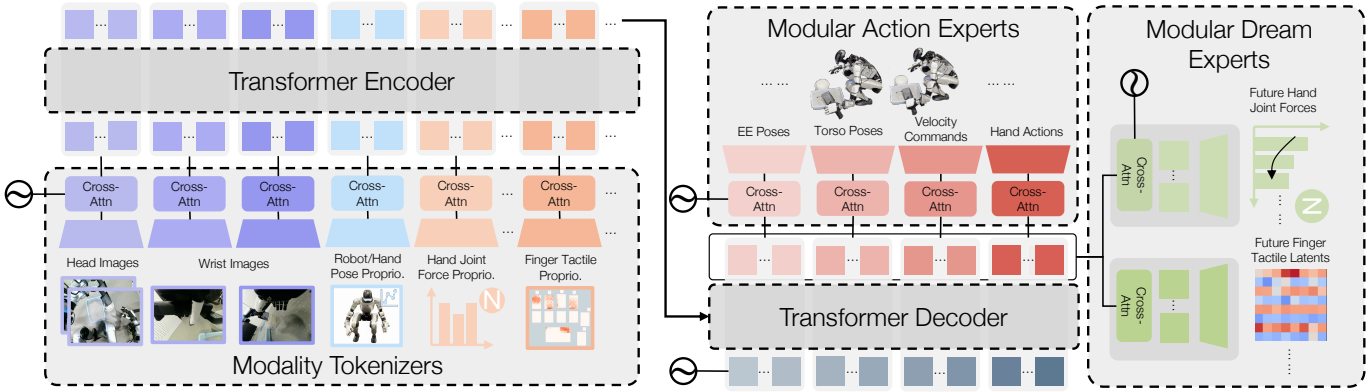


Fig. 4: **HTD model architecture.** HTD is a modular encoder–decoder Transformer. **Left:** modality tokenizers encode multi-view images, proprioception, hand joint forces, and tactile signals into a fixed number of tokens via cross-attention aggregation. **Middle:** a Transformer encoder fuses multimodal observation tokens, and a Transformer decoder produces a fixed set of output tokens. **Right:** modular *action experts* decode pose/velocity/hand-action targets, while modular *dream experts* predict future forces and tactile latents for touch dreaming. We use learnable query embeddings to flexibly determine how many tokens to use for each input/output modality.

both control actions and auxiliary touch-dreaming predictions. Concretely, given observations including multi-view vision, robot and hand proprioception, hand joint force signals, and tactile readings, the tokenizers jointly produce a sequence of tokens that is fused by the transformer encoder. The transformer decoder then emits a fixed set of output tokens, with each action modality assigned a fixed number of tokens. These tokens are consumed by two families of heads: *action experts* that predict structured action targets for whole-body control, and *dream experts* that predict future touch signals (forces and tactile latents) for touch dreaming. The *dream experts* attend to the full set of output tokens across all action modalities. This design incentives the latent dynamics of the shared transformer trunk to be contact-aware.

Modality Tokenizers. Each tokenizer T_m maps a raw modality into a fixed number of tokens, which are concatenated in a fixed order to form the transformer encoder input, similar to [36, 42]. As illustrated in Fig. 4 (left), we first extract modality-specific features, then compress them into tokens using a cross-attention aggregation layer, where a small set of learnable query ("slot") tokens attends to the feature sequence. For image modalities, we extract features with a pretrained ResNet [13] backbone (finetuned during training) and use separate tokenizers for the head camera and each wrist camera. For state-like modalities (e.g., robot/hand pose and proprioception, as well as force-related proprioceptive signals), we use lightweight MLP feature extractors. For tactile inputs, we use a dedicated tactile encoder to embed the raw tactile readings into a compact feature sequence, which is then tokenized via the same cross-attention aggregation.

Per-Finger/Region Tactile Encoder. For tactile inputs, we encode each finger or hand region independently rather than forming a single full-hand tactile embedding upfront. Concretely, the tactile observations are decomposed into anatomically defined inputs corresponding to the thumb, index finger, middle finger, ring finger, pinky, and palm. For a regular finger, the 185-dimensional tactile input is further segmented into three local patches (tip, top, and palm-facing region); for the thumb, the 210-dimensional input is segmented into four patches (tip, top, mid, and palm-facing region); and for the palm, the 112-dimensional input is treated as a single large patch. Each local patch is reshaped into a 2D map and processed by a dedicated CNN branch selected according to patch size, with lightweight single-layer convolutions for small patches and deeper two-layer CNN blocks for larger patches. The resulting patch features are adaptively pooled to a fixed spatial resolution, flattened, concatenated, and fused by an MLP into a compact embedding for that finger or region. These per-region embeddings are then projected to the Transformer hidden dimension and converted into tactile tokens through the same cross-attention aggregation used by the other modality tokenizers. The same per-region tactile encoder architecture is also used to instantiate the EMA target encoder for stable latent supervision during touch dreaming.

Transformer Trunk. HTD uses an encoder–decoder transformer trunk with fixed input and output sequence lengths determined by the number of tokens allocated to each modality and each output group. The encoder contextualizes the concatenated observation tokens into a unified representation. The decoder produces a fixed set of output tokens at prespecified positions. The learnable query embeddings serve as a structured interface that supports multiple downstream experts. This separation enables the encoder to focus on multimodal state understanding while the decoder provides disentangled readouts for control and prediction.

Modular Action Experts. We decode control outputs with a set of *modular action experts* (Fig. 4, top-right). Each expert uses a cross-attention layer to read from the decoder output tokens and predicts a particular action modality, including end-effector pose targets, torso pose targets, velocity commands (when applicable), and hand actions. This modular design allows action modalities with different dimensionalities and control roles to be read out independently and adaptively. In particular, each action modality is assigned its own fixed

Algorithm 1 Imitation learning with touch dreaming

Input: Dataset $\mathcal{D}$ with tuples $(\boldsymbol{o}_t, A_t, F_{t:t+\tau}, S_{t:t+\tau})$
Input: Action chunk horizon h, touch dreaming horizon τ
Input: EMA decay α, loss weights λ_F, λ_Z, magnitude weight β, learning rate η
Output: HTD Policy π_Θ

Initialize policy π_Θ (Θ has parameters from tokenizers, encoder–decoder trunk, and detokenizers; $\theta \in \Theta$)
Initialize teacher tactile tokenizer, $\theta^T = \theta$
for step $= 1, 2, \ldots$ **do**
  Sample a batch $B = \{(\boldsymbol{o}_j, A_j, F_j, S_j)\}_{j=1}^n$ from $\mathcal{D}$
  Teacher latents: compute $\boldsymbol{z}^\star_{j,k}$ with Eq. (8)
  Policy rollout: $\hat{A}_j, \{\hat{\boldsymbol{f}}_{j,k}\}_{k=1}^\tau, \{\hat{\boldsymbol{z}}_{j,k}\}_{k=1}^\tau \sim \pi_\Theta(\boldsymbol{o}_j)$
  Compute total loss $\mathcal{L}(B;\Theta)$ with Eq. (5)
  Update student: $\Theta^S \leftarrow \Theta^S - \eta \nabla_{\Theta^S} \mathcal{L}(B;\Theta)$
  Update EMA teacher: $\theta^T \leftarrow \alpha\theta^T + (1-\alpha)\theta^S$
end for
return π_Θ

number of decoder output tokens, so low-dimensional but behaviorally important outputs such as velocity commands can still receive sufficient representational capacity, while higher-dimensional outputs such as pose or hand-action targets can be decoded by separate experts matched to their complexity. We adopt action chunking [57], where each expert predicts a short horizon of targets at each inference step.

Modular Dream Experts and Touch Dreaming. In addition to action experts, HTD includes *modular dream experts* that provide auxiliary prediction objectives during training (Fig. 4, far-right). These experts predict future touch outcomes, including (i) future hand joint force vectors and (ii) future finger/region tactile *latents*. We refer to these auxiliary predictions as *touch dreaming*: conditioned on the current multimodal observations, the model "imagines" near-future touch feedback, which regularizes the shared transformer trunk to learn contact-aware representations. Crucially, for tactile we perform prediction in a learned latent space rather than raw sensor space. Direct regression in raw tactile space is often dominated by sparsity and noise, whereas latent supervision provides a compact target that captures contact structure. We obtain stable latent labels using an EMA tactile tokenizer as teacher (Sec. III-E), and supervise the student to match these teacher latents. During deployment, only the action experts are used for control; dream experts' outputs are not used.

Modality Decomposition. We preserve semantically distinct inputs as separate modalities and tokenize them independently. On the output side, we similarly decode different action modalities with separate action experts, and decode touch-dreaming targets with dedicated dream experts. Given the distinct statistics of different input/output modalities, this network design allows for modality-based specialization, while the shared transformer trunk learns unified representation and models complex dynamics.

E. Training Paradigm

We train the policy with a single-stage behavioral cloning (BC) paradigm on humanoid demonstrations. The key component of our architecture is *touch dreaming*: in addition to predicting action chunks, the model is trained to predict future touch signals. Specifically, we (i) predict *future hand joint force vectors* using a smooth L1 loss, and (ii) predict *future tactile latents* in a stable latent space generated by an EMA teacher encoder. We find that supervising tactile predictions in latent space instead of regressing raw tactile arrays yields substantially better manipulation performance on real robots, because it provides a compact and semantically rich learning signal while avoiding the difficulty of reconstructing sparse, high-dimensional sensor readings. This auxiliary predictive objective encourages the Transformer trunk to learn contact-aware world representations that transfer to improved downstream contact-rich manipulation. The overall training procedure is summarized in Algorithm 1.

EMA Teacher for Tactile Latents. Let $T_{\text{tact}}(\theta)$ denote the student tactile tokenizer parameterized by θ and $T^T_{\text{tact}}(\theta^T)$ its EMA counterpart. After each optimization step, teacher parameters are updated as an EMA of the student parameters:

$$\theta^T \leftarrow \alpha\theta^T + (1-\alpha)\theta, \qquad \alpha \in (0,1), \tag{4}$$

and no gradient is backpropagated through the teacher. The teacher network provides slowly evolving, temporally consistent latent targets. Without such as self-distillation mechanism, the student tactile tokenizer and the touch detokenizer will mode collapse where all tactile inputs map to near-identical latents regardless of actual contact state.

Objective. Let dataset $\mathcal{D} = \{(\boldsymbol{o}_t, A_t, F_{t:t+\tau}, S_{t:t+\tau})\}$, where $\boldsymbol{o}_t$ is the multimodal observation at time t, $A_t = \{\boldsymbol{a}_{t+\ell}\}_{\ell=1}^h$ is the action chunk of horizon h, $F_{t:t+\tau} = \{\boldsymbol{f}_{t+\ell}\}_{\ell=1}^\tau$ is the future hand joint force sequence and $S_{t:t+\tau} = \{\boldsymbol{s}_{t+\ell}\}_{\ell=1}^\tau$ is the future tactile signal sequence over a prediction horizon τ. Given action modalities $o_1, \ldots, m_K$ and touch signals (force and tactile), the overall loss is:

$$\mathcal{L}(\Theta) = \underbrace{\sum_{i=1}^{K} \mathcal{L}_{\text{act, m}_\text{i}}(\Theta)}_{\text{behavior cloning}} + \lambda_F \underbrace{\mathcal{L}_{\text{force}}(\Theta)}_{\text{force prediction}} + \lambda_Z \underbrace{\mathcal{L}_{\text{tact}}(\Theta)}_{\text{tactile latent prediction}}, \tag{5}$$

where λ_F and λ_Z weight the touch-related objectives. For a batch $B = \{(\boldsymbol{o}_j, A_j, F_j, S_j)\}_{j=1}^n$, the BC loss with action chunking is:

$$\mathcal{L}_{\text{act, m}_\text{i}}(B) = \frac{1}{n}\sum_{j=1}^{n}\left[\frac{1}{h}\sum_{\ell=1}^{h} \ell_1(\boldsymbol{a}_{j,\ell}[m_i], \hat{\boldsymbol{a}}_{j,\ell}[m_i])\right], \tag{6}$$

where $\hat{\boldsymbol{a}}_{j,\ell} = [\pi_\theta(\boldsymbol{o}_j)]_\ell$ denotes the ℓ-th action in the chunk.

Future Hand Joint Force Prediction Loss. For force dreaming, the model predicts future force vectors $\hat{\boldsymbol{f}}_{j,k}$ for $k \in \{1, \ldots, \tau\}$, supervised with the same smooth L1 loss used

for action prediction:

$$\mathcal{L}_{\text{force}}(B) = \frac{1}{n}\sum_{j=1}^{n}\left[\frac{1}{\tau}\sum_{k=1}^{\tau}\ell_1\Big(\hat{\boldsymbol{f}}_{j,k},\, \boldsymbol{f}_{j,k}\Big)\right]. \tag{7}$$

Tactile Dreaming Loss (Latent Supervision). For tactile dreaming, we supervise the model to predict *future tactile latents* rather than raw tactile heatmaps. For each future step $k \in \{1, \dots, \tau\}$, we compute target latent labels by encoding future tactile measurements with the EMA teacher encoder:

$$\boldsymbol{z}^{\star}_{j,k} = \text{stopgrad}\big(T^{T}_{\text{tact}}(\boldsymbol{s}_{j,k})\big)\,, \tag{8}$$

and the touch detokenizer predicts $\hat{\boldsymbol{z}}_{j,k}$. We combine a cosine direction loss with a magnitude alignment loss:

$$\mathcal{L}_{\text{tact}}(B) = \frac{1}{n}\sum_{j=1}^{n}\left[\frac{1}{\tau}\sum_{k=1}^{\tau}\Big(\underbrace{1-\cos(\hat{\boldsymbol{z}}_{j,k},\, \boldsymbol{z}^{\star}_{j,k})}_{\text{direction}} + \beta\,\underbrace{\ell_{\delta}\big(\|\hat{\boldsymbol{z}}_{j,k}\| - \|\boldsymbol{z}^{\star}_{j,k}\|\big)}_{\text{magnitude}}\Big)\right], \tag{9}$$

where $\cos(\cdot,\cdot)$ denotes cosine similarity, ℓ_δ is the smooth L1 loss, and β controls the relative weight of magnitude alignment. The direction term encourages the predicted latent to align with the teacher target in orientation, while the magnitude term ensures the predicted norm matches, preventing the model from collapsing to unit-norm predictions that satisfy cosine similarity alone.

IV. Experiments

We conduct experiments to answer the following research questions: 1) How does our LBC strategy compare to other methods with regards to tracking accuracy and robustness? 2) How do our learning system and HTD perform on real–world versatile humanoid manipulation? 3) How do predictive touch dreaming and latent tactile supervision impact the effectiveness of contact-rich humanoid manipulation? 4) Does touch dreaming capture meaningful and robust contact-aware representations?

A. How does our LBC strategy compare to other methods with regards to tracking accuracy and robustness?

To evaluate tracking performance, we benchmark our LBC against two representative decoupled whole-body humanoid controllers: FALCON [55], which uses a dual-policy architecture with separate upper- and lower-body control and adaptive force curriculum learning; and AMO [25], which uses a hierarchical whole-body control framework where upper-body goals and torso commands are converted into lower-body references through an adaptive motion optimization module and tracked by a learned lower policy. We quantify performance with tracking error metrics:

- Linear Velocity Tracking Error E_v: measures the L2 error between commanded and actual forward/lateral velocities in the robot's yaw-aligned horizontal frame.

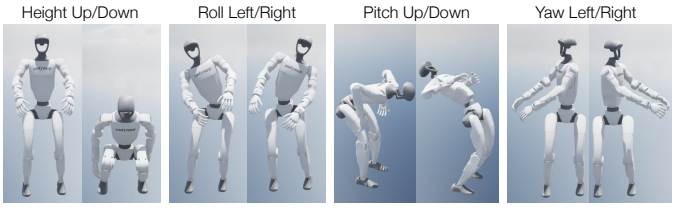


Fig. 5: Visualization of postures near the boundary of the stable controllable workspace of our LBC policy in simulation.

- Height Tracking Error E_h: quantifies the deviation of torso height from the commanded value.
- Yaw Orientation Tracking Error E_y: evaluates the error in relative yaw angle between torso and pelvis.
- Pitch Orientation Tracking Error E_p: measures the absolute pitch angle deviation of the torso from the commanded upright orientation.
- Roll Orientation Tracking Error E_r: assesses the error in relative roll angle between torso and pelvis.

All orientations are computed using intrinsic XYZ Euler decomposition. For evaluation, we run each baseline using their publicly available implementations across 4096 parallel simulation environments for 500 timesteps. We calculate metrics by averaging tracking errors over all timesteps and environments, providing statistically robust assessment of sustained tracking performance across varied conditions.

Our LBC achieves the best overall tracking on most metrics in Table II, notably linear velocity and torso height and orientation (E_h, E_y, E_p, E_r), indicating tighter torso pose regulation that is crucial for contact-rich whole-body manipulation. While AMO attains a slightly lower yaw rate error E_ω, our controller provides a stronger balance between locomotion tracking and whole-body configuration control. We note that the standard deviations of E_h and especially E_p exceed their means, largely due to a small number of difficult command combinations that create conflicting objectives, e.g., simultaneously requesting large forward torso pitch and a low base height; in these cases the policy may temporarily trade off tracking for stability, increasing variance despite low average errors across the command space.

TABLE II: Tracking Error Comparison

Metric	Ours	AMO [25]	FALCON [55]
E_v (m/s)	**0.1420 ± 0.0568**	0.1779 ± 0.0642	0.1641 ± 0.0309
E_ω (rad/s)	0.1806 ± 0.0534	**0.1540 ± 0.0316**	0.1874 ± 0.0263
E_h (m)	**0.0280 ± 0.0438**	0.0568 ± 0.0814	0.1299 ± 0.0082
E_y (rad)	**0.0126 ± 0.0051**	0.1540 ± 0.0534	0.1215 ± 0.0111
E_p (rad)	**0.0487 ± 0.1796**	0.1519 ± 0.1254	*(not tracked)*
E_r (rad)	**0.0157 ± 0.0065**	0.0735 ± 0.0447	*(not tracked)*

Average tracking error alone does not fully characterize the usable operating envelope of a whole-body controller, since manipulation-oriented control requires stability under large torso reorientation and height variation that expand reachability. We therefore measure the per-dimension stable controllable range of our LBC policy in simulation by sweeping one command dimension at a time, base height h, torso roll ϕ_{torso}, torso pitch θ_{torso}, and torso yaw ψ_{torso}, while keeping the others nominal, and running the policy

until instability, a fall, or persistent tracking failure occurs; we report the largest interval that remains stable over the rollout. Our policy achieves stable command-tracking ranges of $h \in [0.33, 0.80]$ m, $\phi_{\text{torso}} \in [-0.38, 0.35]$ rad, $\theta_{\text{torso}} \in [-0.92, 1.41]$ rad, and $\psi_{\text{torso}} \in [-1.50, 1.34]$ rad. The stable workspace covers the trained height range and most of the yaw range, while roll remains most restrictive. Fig. 5 visualizes boundary postures, showing that our controller supports a wide stable region for crouching, bending, and large torso reorientation beyond nominal upright motions.

B. How do our learning system and HTD perform on real-world versatile humanoid manipulation?

To answer this question, we evaluate our learning system on five contact-rich humanoid manipulation tasks:

- **Insert-T:** The robot grasps a T-shaped block randomly initialized within a region on the table and inserts it into a fixed T-shaped base with a tight clearance of 3.5 mm. This task tests contact-sensitive correction and high spatial accuracy under small insertion tolerances.
- **Book Organization:** The robot gently pushes a hardcover book, chosen from two variants and randomly initialized within a region on the table, to create a graspable overhang, since the book is difficult to pick up directly from a flat surface. It then securely grasps the book and places it onto a bookshelf. This task tests hybrid pushing-and-grasping and controlled reorientation and placement for thin rigid objects with limited grasp affordance.
- **Towel Folding:** The robot folds a towel on the table. The towel is randomly initialized within a region on the table, with three different initial folding configurations. This task tests deformable object handling over multiple manipulation stages.
- **Cat Litter Scooping:** The robot squats to reach a litter scoop on the ground, uses it to scoop 3D-printed litter from a box, and dumps the litter into a trash bin. The litter is randomly distributed within the box, the trash bin is randomly placed on the right side of the box, and the scoop is randomly placed near the left edge of the box with two pose variations. This task tests tool-mediated interaction and whole-body reachability.
- **Tea Serving:** The robot walks to a bar, picks up two cups of tea randomly positioned within a region on the bar table, carries them to a nearby table, comes to a stop, and places both cups on the tabletop. This task tests dual-arm loco-manipulation to maintain object stability throughout the motion.

Baselines. We compare our approach against two decoder-only variants of ACT [57], which we found empirically perform better than the version augmented with a CVAE encoder. **ACT (Visual + Proprio)** uses only multi-view vision and proprioception. **ACT (Visual + Proprio + Touch)** additionally takes force and tactile observations as input. We report results for our method, **HTD**, which augments imitation learning with touch dreaming and employs a modular design for observations, action experts, and dream experts.

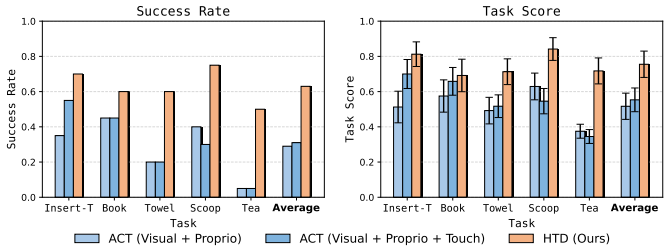


Fig. 6: Real-world results on five contact-rich tasks. We compare ACT (Visual + Proprio), ACT (Visual + Proprio + Touch), and HTD. Left: success rate. Right: score rate (mean $\pm$ SEM, 20 trials).

Metrics and protocol. For each task and method, we run 20 real-world trials and report the score rate (mean $\pm$ SEM) and success rate. Score rate reflects task completion quality under partial progress; success rate measures strict task completion.

Results and analysis. Fig. 6 summarizes the main results. Across all five tasks, HTD consistently outperforms both decoder-only ACT baselines in both success rate and task score. Averaged across tasks, HTD improves over the stronger ACT variant by 30.0 percentage points in success rate and 17.9 percentage points in task score, corresponding to relative gains of about 90.9% and 31.1%, respectively. Importantly, simply adding force and tactile observations to ACT does not consistently improve performance: ACT (Visual + Proprio + Touch) outperforms ACT (Visual + Proprio) on only a subset of tasks and is not uniformly better in either metric. The largest gains of HTD appear on tasks that place stronger demands on contact-aware control or whole-body coordination. *Insert-T* benefits from more accurate handling of tight-tolerance alignment and corrective contact. *Towel Folding* highlights HTD's advantage on long-horizon deformable manipulation. *Cat Litter Scooping* shows particularly large gains, reflecting the difficulty of combining tool use with squatting and constrained whole-body motion. In *Tea Serving*, ACT often fails to rotate and move the body appropriately after successfully grasping both tea cups, whereas HTD is much more reliable. This likely reflects the importance of decoding low-dimensional but behavior-critical velocity commands with dedicated output tokens and independent action experts, rather than treating them as a small subset of a monolithic action vector. *Book Organization* shows a smaller but still consistent gain, likely because it is more visually structured and has lower object-location variance. Overall, these results indicate that the full HTD framework is better suited than ACT baselines for versatile humanoid loco-manipulation.

C. How do predictive touch dreaming and latent tactile supervision impact the effectiveness of contact-rich humanoid manipulation?

To answer this question, we further isolate the contribution of touch observations and touch dreaming via ablations of HTD. Specifically, we evaluate four variants: **w/o Touch and TD** removes both touch observations and the touch dreaming objective. **w/o TD** keeps touch observations as input but removes the dreaming loss. **Dream Raw Tactile** predicts

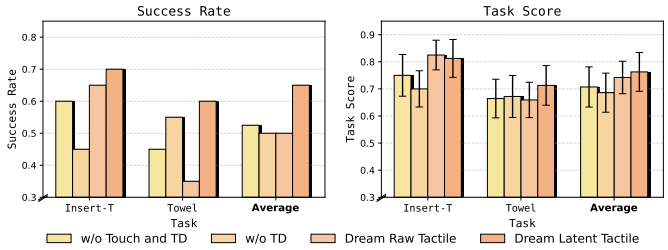


Fig. 7: Ablations of HTD. Variants: w/o Touch and TD, w/o TD, Dream Raw Tactile, and Dream Latent Tactile (full). Left: success rate. Right: score rate (mean ± SEM, 20 trials).

future tactile signals in the raw sensor space. **Dream Latent Tactile** (our full method) predicts future tactile signals in a learned latent space supervised by an EMA teacher. All variants are trained with the same behavioral cloning objective for action chunk prediction, differing only in touch-related inputs and auxiliary objectives.

Results and Analysis. Fig. 7 reveals three main observations. First, *touch as an input alone is not consistently beneficial*. Comparing w/o Touch and TD with w/o TD, adding touch observations without touch dreaming improves performance on Towel Folding, but does not help on Insert-T, and is slightly worse on average in success rate. The task-score differences between the two variants are also small. This indicates that simply appending touch observations does not reliably translate into better control performance.

Second, adding a predictive touch objective improves performance beyond passive touch conditioning. Both Dream Raw Tactile and Dream Latent Tactile outperform w/o TD on *Insert-T*, *Towel Folding*, and on the average metrics, showing that explicitly learning to anticipate future contact provides a more useful training signal than using current touch observations alone.

Third, Dream Latent Tactile achieves the best overall performance and consistently outperforms Dream Raw Tactile, especially in success rate, where it yields a relative gain of 30%. This suggests that supervising future touch in a learned latent space is more effective and stable than directly predicting raw tactile signals. Overall, these ablations support two conclusions: simply adding touch inputs is insufficient, and the choice of supervision for future touch has a substantial impact on downstream contact-rich humanoid manipulation.

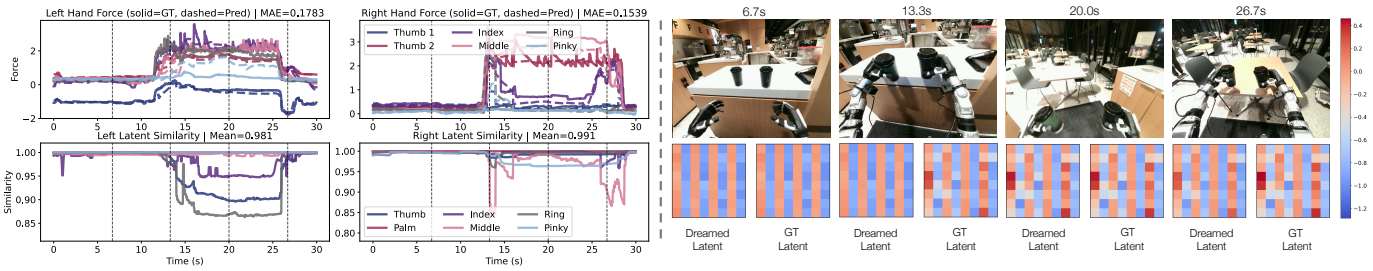
D. Does touch dreaming capture meaningful and robust contact-aware representations?

Qualitative touch dreaming. To better understand what the touch-dreaming objective learns, we qualitatively examine whether the predicted future forces and tactile latents reflect meaningful contact dynamics during real-world rollouts. Fig. 8 illustrates the dreamed touch representations across two policy rollout episodes, *Tea Serving* and *Towel Folding*. Across these representative rollouts, HTD predicts future hand force trajectories that effectively track both the timing and magnitude of contact events. Furthermore, tactile latent similarity (L2 similarity) remains high during periods of sustained contact. On the right side of Fig. 8, we show the synchronized right-eye camera view together with heatmaps of the raw tactile latent values for the selected fingers at the timesteps indicated by the dashed vertical lines in the plots on the left. We observe that the predicted heatmaps are close to the ground-truth heatmaps at most timesteps. While temporary drops in similarity occur during abrupt contact transitions and coincide with sudden force spikes, the similarity remains relatively high overall. The localized deviations are expected for two reasons. First, because the dreamed latent chunks are rolled out in an open-loop manner, the predictions naturally diverge slightly from the ground truth when unpredictable, discontinuous contact changes occur mid-chunk. Second, the ground-truth tactile latents typically vary with the noisy raw tactile readings, whereas the dreamed tactile latents are more stable and more aligned with semantically meaningful contact changes inferred jointly from vision and applied forces, rather than directly matching raw tactile readings that can be **noisy** or **weakly responsive**.

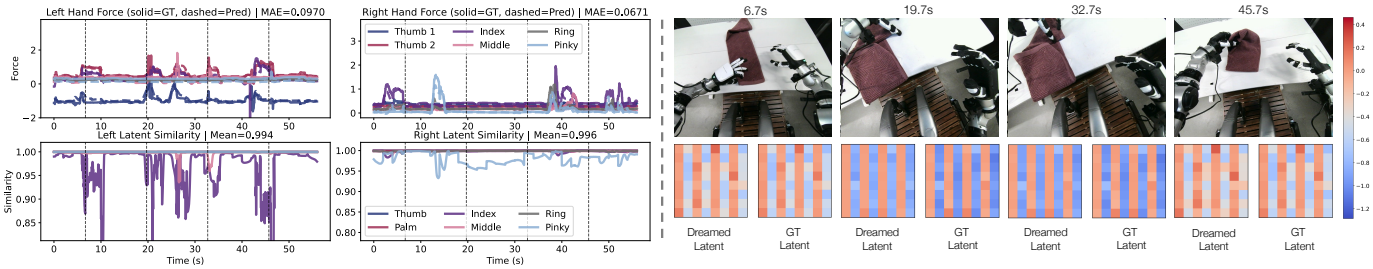
Comparing the two tasks further highlights the contact-awareness of the learned representations. In general, *Tea Serving* involves rigid objects and requires larger applied forces than the deformable interaction in *Towel Folding*. During phases with light or sparse contact, such as *Tea Serving* at 6.7s or the preliminary contacts in *Towel Folding* at 19.7s and 32.7s, the latent patterns remain highly consistent across different fingers and tasks. Conversely, when richer contact occurs, the latents activate into distinct, high-intensity patterns, as seen in *Tea Serving* at 13.3s, 20.0s, and 26.7s, and in *Towel Folding* at 6.7s and 45.7s. Notably, when fingers experience comparable contact states, such as similar force magnitudes and localized contact regions, the resulting tactile latents exhibit visually analogous structural patterns. Meanwhile, the predicted tactile latents also show larger peak values under stronger contact and larger applied forces, for example when comparing the active contact patterns in *Tea Serving* with those in *Towel Folding*. This consistency across varying contact intensities suggests that the learned tactile latent space captures contact-relevant structure while being less affected by the spatial sparsity and local irregularity typical of raw sensor signals. Overall, these qualitative results reinforce the quantitative ablations by showing that the model learns a robust and contact-aware latent representation of physical interaction.

V. Conclusions

In this work, we study *dexterous, contact-rich humanoid manipulation* and present an integrated system that combines a robust lower-body controller for stable whole-body execution, a real-world VR-based data collection pipeline for humanoid demonstrations, and a touch-aware policy learning framework. We propose **Humanoid Transformer with Touch Dreaming (HTD)**, a multimodal encoder–decoder Transformer that models touch as a core modality and augments single-stage behavioral cloning with future hand-joint-force prediction and future tactile-latent prediction. By supervising future tactile prediction with an EMA target encoder, HTD learns stable,



(a) **Tea Serving.** Heatmaps with raw latent values correspond to the **right middle** finger.



(b) **Towel Folding.** Heatmaps with raw latent values correspond to the **left index** finger.

Fig. 8: **Touch dreaming visualization.** We compare predicted (Pred) versus ground-truth (GT) future contact signals on representative rollouts for two tasks. For each task, the top left shows per-finger hand force trajectories and the mean absolute error (MAE) for the left and right hands, and the bottom left shows the corresponding tactile latent similarity over time (computed with L2 similarity). The vertical dashed lines in the left plots indicate the specific timestamps for the synchronized camera views and heatmaps of the dreamed versus ground-truth tactile latents shown on the right.

contact-aware latent representations without requiring a separate tactile pretraining stage.

Across five real-world tasks, *Insert-T*, *Book Organization*, *Towel Folding*, *Cat Litter Scooping*, and *Tea Serving*, HTD consistently outperforms decoder-only ACT baselines, achieving **90.9%** relative improvement in average success rate over the stronger ACT variant. Our ablations further show that latent tactile dreaming is more effective than raw tactile prediction, with Dream Latent Tactile yielding a **30%** relative gain in success rate over Dream Raw Tactile. Together, these results suggest that combining robust whole-body execution, scalable humanoid data collection, and predictive touch-centered learning provides a practical path toward more reliable humanoid manipulation under frequent and complex contact changes.

References

[1] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15619–15629, 2023.

[2] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

[3] Qingwei Ben, Feiyu Jia, Jia Zeng, Junting Dong, Dahua Lin, and Jiangmiao Pang. Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit. *arXiv preprint arXiv:2502.13013*, 2025.

[4] Jianxin Bi, Kevin Yuchen Ma, Ce Hao, Mike Zheng Shou, and Harold Soh. Vla-touch: Enhancing vision-language-action models with dual-level tactile feedback. *arXiv preprint arXiv:2507.17294*, 2025.

[5] Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H Adelson, and Sergey Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307, 2018.

[6] Haonan Chen, Jiaming Xu, Hongyu Chen, Kaiwen Hong, Binghao Huang, Chaoqi Liu, Jiayuan Mao, Yunzhu Li, Yilun Du, and Katherine Driggs-Campbell. Multi-modal manipulation via multi-modal policy consensus. *arXiv preprint arXiv:2509.23468*, 2025.

[7] Sirui Chen, Zi-ang Cao, Zhengyi Luo, Fernando Castañeda, Chenran Li, Tingwu Wang, Ye Yuan, Linxi Fan, C Karen Liu, Yuke Zhu, et al. Chip: Adaptive compliance for humanoid control through hindsight perturbation. *arXiv preprint arXiv:2512.14689*, 2025.

[8] Wendi Chen, Han Xue, Yi Wang, Fangyuan Zhou, Jun Lv, Yang Jin, Shirun Tang, Chuan Wen, and Cewu Lu. Implicitrdp: An end-to-end visual-force diffusion policy with structural slow-fast learning. *arXiv preprint arXiv:2512.10946*, 2025.

[9] Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive whole-body control for humanoid robots. *arXiv preprint*

arXiv:2402.16796, 2024.

[10] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. *arXiv preprint arXiv:2407.01512*, 2024.

[11] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.

[12] Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9164–9170. IEEE, 2020.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

[15] Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning human-to-humanoid real-time whole-body teleoperation. *arXiv preprint arXiv:2403.04436*, 2024.

[16] Erik Helmut, Niklas Funk, Tim Schneider, Cristiana de Farias, and Jan Peters. Tactile-conditioned diffusion policy for force-aware robotic manipulation. *arXiv preprint arXiv:2510.13324*, 2025.

[17] Liang Heng, Haoran Geng, Kaifeng Zhang, Pieter Abbeel, and Jitendra Malik. Vitacformer: Learning cross-modal representation for visuo-tactile dexterous manipulation. *arXiv preprint arXiv:2506.15953*, 2025.

[18] Liang Heng, Yihe Tang, Jiajun Xu, Henghui Bao, Di Huang, and Yue Wang. Humdex: Humanoid dexterous manipulation made easy. *arXiv preprint arXiv:2603.12260*, 2026.

[19] Carolina Higuera, Akash Sharma, Chaithanya Krishna Bodduluri, Taosha Fan, Patrick Lancaster, Mrinal Kalakrishnan, Michael Kaess, Byron Boots, Mike Lambeta, Tingfan Wu, and Mustafa Mukadam. Sparsh: Self-supervised touch representations for vision-based tactile sensing. 2024. URL https://openreview.net/forum?id=xYJn2e1uu8.

[20] Carolina Higuera, Sergio Arnaud, Byron Boots, Mustafa Mukadam, Francois Robert Hogan, and Franziska Meier. Visuo-tactile world models. *arXiv preprint arXiv:2602.06001*, 2026.

[21] Binghao Huang, Yixuan Wang, Xinyi Yang, Yiyue Luo, and Yunzhu Li. 3d-vitac: Learning fine-grained manipulation with visuo-tactile sensing. *arXiv preprint arXiv:2410.24091*, 2024.

[22] Jialei Huang, Shuo Wang, Fanqi Lin, Yihang Hu, Chuan Wen, and Yang Gao. Tactile-vla: unlocking vision-language-action model's physical knowledge for tactile generalization. *arXiv preprint arXiv:2507.09160*, 2025.

[23] Eunju Kwon, Seungwon Oh, In-Chang Baek, Yucheon Park, Gyungbo Kim, JaeYoung Moon, Yunho Choi, and Kyung-Joong Kim. A humanoid visual-tactile-action dataset for contact-rich manipulation. *arXiv preprint arXiv:2510.25725*, 2025.

[24] Michelle A Lee, Yuke Zhu, Peter Zachares, Matthew Tan, Krishnan Srinivasan, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Learning multimodal representations for contact-rich tasks. *IEEE Transactions on Robotics*, 36(3):582–596, 2020.

[25] Jialong Li, Xuxin Cheng, Tianshu Huang, Shiqi Yang, Ri-Zhao Qiu, and Xiaolong Wang. Amo: Adaptive motion optimization for hyper-dexterous humanoid whole-body control. *arXiv preprint arXiv:2505.03738*, 2025.

[26] 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. *arXiv preprint arXiv:2410.11792*, 2024.

[27] Yixuan Li, Yutang Lin, Jieming Cui, Tengyu Liu, Wei Liang, Yixin Zhu, and Siyuan Huang. Clone: Closed-loop whole-body humanoid teleoperation for long-horizon tasks. In *9th Annual Conference on Robot Learning*, 2025.

[28] Yixuan Li, Le Ma, Yutang Lin, Yushi Du, Mengya Liu, Kaizhe Hu, Jieming Cui, Yixin Zhu, Wei Liang, Baoxiong Jia, et al. Omniclone: Engineering a robust, all-rounder whole-body humanoid teleoperation system. *arXiv preprint arXiv:2603.14327*, 2026.

[29] Qiayuan Liao, Takara E Truong, Xiaoyu Huang, Yuman Gao, Guy Tevet, Koushil Sreenath, and C Karen Liu. Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion. *arXiv preprint arXiv:2508.08241*, 2025.

[30] Toru Lin, Yu Zhang, Qiyang Li, Haozhi Qi, Brent Yi, Sergey Levine, and Jitendra Malik. Learning visuotactile skills with two multifingered hands. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5637–5643. IEEE, 2025.

[31] Chenhao Lu, Xuxin Cheng, Jialong Li, Shiqi Yang, Mazeyu Ji, Chengjing Yuan, Ge Yang, Sha Yi, and Xiaolong Wang. Mobile-television: Predictive motion priors for humanoid whole-body control. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5364–5371. IEEE, 2025.

[32] Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Sirui Chen, Fernando Castaneda, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, et al. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *arXiv preprint arXiv:2511.07820*, 2025.

[33] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings*

of the IEEE/CVF international conference on computer vision, pages 5442–5451, 2019.

[34] Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Muñoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

[35] Ruiqian Nai, Boyuan Zheng, Junming Zhao, Haodong Zhu, Sicong Dai, Zunhao Chen, Yihang Hu, Yingdong Hu, Tong Zhang, Chuan Wen, and Yang Gao. Humanoid manipulation interface: Humanoid whole-body manipulation from robot-free demonstrations, 2026. URL https://arxiv.org/abs/2602.06643.

[36] Yaru Niu, Yunzhe Zhang, Mingyang Yu, Changyi Lin, Chenhao Li, Yikai Wang, Yuxiang Yang, Wenhao Yu, Tingnan Zhang, Zhenzhen Li, Jonathan Francis, Bingqing Chen, Jie Tan, and Ding Zhao. Human2locoman: Learning versatile quadrupedal manipulation with human pretraining. In *Robotics: Science and Systems (RSS)*, 2025.

[37] Haozhi Qi, Yen-Jen Wang, Toru Lin, Brent Yi, Yi Ma, Koushil Sreenath, and Jitendra Malik. Coordinated humanoid manipulation with choice policies. *arXiv preprint arXiv:2512.25072*, 2025.

[38] Ri-Zhao Qiu, Shiqi Yang, Xuxin Cheng, Chaitanya Chawla, Jialong Li, Tairan He, Ge Yan, David J Yoon, Ryan Hoque, Lars Paulsen, et al. Humanoid policy˜ human policy. *arXiv preprint arXiv:2503.13441*, 2025.

[39] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

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

[41] Wandong Sun, Luying Feng, Baoshi Cao, Yang Liu, Yaochu Jin, and Zongwu Xie. Ulc: A unified and fine-grained controller for humanoid loco-manipulation, 2025. URL https://arxiv.org/abs/2507.06905.

[42] Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. *Advances in neural information processing systems*, 37:124420–124450, 2024.

[43] Lai Wei, Xuanbin Peng, Ri-Zhao Qiu, Tianshu Huang, Xuxin Cheng, and Xiaolong Wang. Hmc: Learning heterogeneous meta-control for contact-rich loco-manipulation. *arXiv preprint arXiv:2511.14756*, 2025.

[44] Zhen Wu, Xiaoyu Huang, Lujie Yang, Yuanhang Zhang, Koushil Sreenath, Xi Chen, Pieter Abbeel, Rocky Duan, Angjoo Kanazawa, Carmelo Sferrazza, et al. Perceptive humanoid parkour: Chaining dynamic human skills via motion matching. *arXiv preprint arXiv:2602.15827*, 2026.

[45] Han Xue, Jieji Ren, Wendi Chen, Gu Zhang, Yuan Fang, Guoying Gu, Huazhe Xu, and Cewu Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.

[46] Lujie Yang, Xiaoyu Huang, Zhen Wu, Angjoo Kanazawa, Pieter Abbeel, Carmelo Sferrazza, C Karen Liu, Rocky Duan, and Guanya Shi. Omniretarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction. *arXiv preprint arXiv:2509.26633*, 2025.

[47] Guo Ye, Zexi Zhang, Xu Zhao, Shang Wu, Haoran Lu, Shihan Lu, and Han Liu. Learning to feel the future: Dreamtacvla for contact-rich manipulation. *arXiv preprint arXiv:2512.23864*, 2025.

[48] Uksang Yoo, Yuemin Mao, Jean Oh, and Jeffrey Ichnowski. A-slip: Acoustic sensing for continuous in-hand slip estimation, 2026. URL https://arxiv.org/abs/2604.08528.

[49] Haoran Yuan, Weigang Yi, Zhenyu Zhang, Wendi Chen, Yuchen Mo, Jiashi Yin, Xinzhuo Li, Xiangyu Zeng, Chuan Wen, Cewu Lu, et al. Vtam: Video-tactile-action models for complex physical interaction beyond vlas. *arXiv preprint arXiv:2603.23481*, 2026.

[50] Yanjie Ze, Zixuan Chen, Joao Pedro Araújo, Zi-ang Cao, Xue Bin Peng, Jiajun Wu, and C Karen Liu. Twist: Teleoperated whole-body imitation system. *arXiv preprint arXiv:2505.02833*, 2025.

[51] Yanjie Ze, Zixuan Chen, Wenhao Wang, Tianyi Chen, Xialin He, Ying Yuan, Xue Bin Peng, and Jiajun Wu. Generalizable humanoid manipulation with 3d diffusion policies. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2873–2880. IEEE, 2025.

[52] Yanjie Ze, Siheng Zhao, Weizhuo Wang, Angjoo Kanazawa, Rocky Duan, Pieter Abbeel, Guanya Shi, Jiajun Wu, and C Karen Liu. Twist2: Scalable, portable, and holistic humanoid data collection system. *arXiv preprint arXiv:2511.02832*, 2025.

[53] Chaofan Zhang, Peng Hao, Xiaoge Cao, Xiaoshuai Hao, Shaowei Cui, and Shuo Wang. Vtla: Vision-tactile-language-action model with preference learning for insertion manipulation. *arXiv preprint arXiv:2505.09577*, 2025.

[54] Xingyu Zhang, Chaofan Zhang, Boyue Zhang, Zhinan Peng, Shaowei Cui, and Shuo Wang. Dextac: Learning contact-aware visuotactile policies via hand-by-hand teaching. *arXiv preprint arXiv:2601.21474*, 2026.

[55] Yuanhang Zhang, Yifu Yuan, Prajwal Gurunath, Ishita Gupta, Shayegan Omidshafiei, Ali-akbar Agha-mohammadi, Marcell Vazquez-Chanlatte, Liam Pedersen, Tairan He, and Guanya Shi. Falcon: Learning force-adaptive humanoid loco-manipulation. *arXiv preprint arXiv:2505.06776*, 2025.

[56] Jialiang Zhao, Yuxiang Ma, Lirui Wang, and Edward H. Adelson. Transferable tactile transformers for represen-

tation learning across diverse sensors and tasks, 2024.

[57] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL https://arxiv.org/abs/2304.13705.

[58] Yuhang Zheng, Songen Gu, Weize Li, Yupeng Zheng, Yujie Zang, Shuai Tian, Xiang Li, Ce Hao, Chen Gao, Si Liu, et al. Omnivta: Visuo-tactile world modeling for contact-rich robotic manipulation. *arXiv preprint arXiv:2603.19201*, 2026.

[59] Tengjie Zhu, Guanyu Cai, Yang Zhaohui, Guanzhu Ren, Haohui Xie, ZiRui Wang, Junsong Wu, Jingbo Wang, Xiaokang Yang, Yao Mu, et al. Clot: Closed-loop global motion tracking for whole-body humanoid teleoperation. *arXiv preprint arXiv:2602.15060*, 2026.

[60] Xinyue Zhu, Binghao Huang, and Yunzhu Li. Touch in the wild: Learning fine-grained manipulation with a portable visuo-tactile gripper. *arXiv preprint arXiv:2507.15062*, 2025.

APPENDIX

A. Lower-Body Controller Details

We provide additional details on the command ranges and domain randomization parameters used in training the lower-body controller.

a) Command ranges and domain randomization: Command signals are uniformly sampled from predefined ranges (Table III), where $\boldsymbol{v} = [v_x, v_y]^T$ denotes the base velocity commands, $\boldsymbol{rpy} = [\phi_{\text{torso}}, \theta_{\text{torso}}, \psi_{\text{torso}}]$ denotes the torso orientation commands, and h is the base height command. To improve sim-to-real transferability, we apply domain randomization across physics parameters (Table IV).

TABLE III: Command ranges

Command	Range	Command	Range
v_x	$[-0.5,\ 0.5]$ m/s	ϕ_{torso} (roll)	$[-0.7,\ 0.7]$ rad
v_y	$[-0.5,\ 0.5]$ m/s	θ_{torso} (pitch)	$[-0.52,\ 1.57]$ rad
ω_z	$[-1.57,\ 1.57]$ rad/s	ψ_{torso} (yaw)	$[-1.57,\ 1.57]$ rad
h	$[0.35,\ 0.8]$ m		

TABLE IV: Domain randomizations

Parameter	Range	Parameter	Range
Angular velocity	± 0.2 rad/s	Static friction	$[0.6,\ 1.0]$
Projected gravity	± 0.05	Dynamic friction	$[0.4,\ 0.8]$
Joint position	± 0.01 rad	Restitution	$[0.0,\ 0.005]$
Joint velocity	± 1.5 rad/s	Base mass	$[-5.0,\ 5.0]$ kg

b) Achievable control ranges: Our learned policy achieves the maximum stable controllable ranges summarized in Table V, where the base height is $[0.33,\ 0.80]$ m, torso roll is $[-0.38,\ 0.35]$ rad, torso pitch is $[-0.92,\ 1.41]$ rad, and torso yaw is $[-1.50,\ 1.34]$ rad. Compared with the command ranges used during training, the controller fully covers the trained height range and even slightly exceeds the lower bound, while covering most of the trained yaw range. For pitch, the stable range is broad but asymmetric: it extends beyond the trained range on the negative side, while being slightly smaller near the positive extreme. In contrast, the roll range is noticeably narrower than the training range, suggesting that lateral whole-body balance remains the most restrictive direction.

TABLE V: Maximum stable controllable ranges of our LBC policy in simulation.

Command	Stable range
Base height h	$[0.33,\ 0.80]$ m
Torso roll ϕ_{torso}	$[-0.38,\ 0.35]$ rad
Torso pitch θ_{torso}	$[-0.92,\ 1.41]$ rad
Torso yaw ψ_{torso}	$[-1.50,\ 1.34]$ rad

B. Reward Details

Table VI summarizes all reward terms used in training. The overall reward is composed of tracking rewards, regularization terms, contact- and gait-related terms, stability terms, and several auxiliary penalties. Below we briefly describe the role of each term.

a) Tracking rewards:

- **Linear velocity reward.** Encourages the robot to track the commanded planar base velocity in the yaw-aligned horizontal frame, which is essential for stable omnidirectional locomotion.
- **Angular velocity reward.** Encourages accurate tracking of the commanded yaw angular velocity, allowing the robot to regulate turning behavior during locomotion.
- **Torso height reward.** Encourages the robot to maintain the desired torso height, which is important for both balance and workspace modulation during manipulation.
- **Torso roll reward.** Encourages tracking of the commanded torso roll relative to the pelvis, enabling lateral torso leaning and posture adjustment.
- **Torso pitch reward.** Encourages tracking of the commanded torso pitch, which is particularly useful for bending motions and forward-reaching manipulation behaviors.
- **Torso yaw reward.** Encourages tracking of the commanded torso yaw relative to the pelvis, allowing the upper body to reorient independently for manipulation and coordination.

b) Regularization terms:

- **Energy penalty.** Penalizes large instantaneous actuation effort to encourage energy-efficient motions and reduce unnecessary torque output.
- **Action rate penalty.** Penalizes rapid changes in consecutive actions, promoting smoother control signals and improving motion consistency.
- **Joint acceleration penalty.** Penalizes large joint accelerations to reduce abrupt motions and encourage physically plausible transitions.
- **Vertical velocity penalty.** Penalizes undesired vertical base motion, helping the robot maintain stable height regulation rather than bouncing during locomotion.
- **Roll/pitch rate penalty.** Penalizes excessive angular velocity around the roll and pitch axes, reducing aggressive body oscillations and improving upper-body stability.

c) Contact and gait terms:

- **Undesired contacts penalty.** Penalizes collisions involving non-foot body parts, encouraging the robot to avoid falling, dragging limbs, or contacting the environment with inappropriate links.
- **Feet slide penalty.** Penalizes foot motion while in contact with the ground, encouraging stable footholds and reducing slip.
- **Flying penalty.** Penalizes states in which no foot is in contact with the ground, discouraging unstable airborne phases during normal locomotion.
- **Feet force penalty.** Penalizes excessively large vertical ground reaction forces, which helps avoid overly harsh stepping and improves motion smoothness.
- **Feet air-time reward.** Encourages appropriate stepping behavior by rewarding meaningful swing phases when non-trivial motion commands are issued.
- **Feet stumble penalty.** Penalizes abnormal contact patterns in which tangential foot force becomes excessively large relative to vertical support force, which often indi-

TABLE VI: Reward terms

Term	Weight	Term	Weight
Tracking Rewards			
Linear velocity $r_{\text{vel}} := \exp\left(-\frac{\lVert \mathbf{v}_{xy} - \mathbf{v}^*_{xy} \rVert_2^2}{\sigma_v^2}\right)$	1.0	**Torso roll** $r_{\text{roll}} := \exp\left(-\frac{(\Delta\phi - \phi^*)^2}{\sigma_r^2}\right)$, $\Delta\phi := \phi_{\text{torso}} - \phi_{\text{pelvis}}$	1.0
Angular velocity $r_{\text{ang}} := \exp\left(-\frac{(\omega_z - \omega_z^*)^2}{\sigma_\omega^2}\right)$	1.0	**Torso pitch** $r_{\text{pitch}} := \exp\left(-\frac{(\theta - \theta^*)^2}{\sigma_p^2}\right)$	1.0
Torso height $r_h := \exp\left(-\frac{(h - h^*)^2}{\sigma_h^2}\right)$	1.0	**Torso yaw** $r_{\text{yaw}} := \exp\left(-\frac{(\Delta\psi - \psi^*)^2}{\sigma_y^2}\right)$, $\Delta\psi := \psi_{\text{torso}} - \psi_{\text{pelvis}}$	1.0
Regularization			
Energy $r_E := \lVert \lvert \boldsymbol{\tau} \odot \dot{\mathbf{q}} \rvert \rVert_2$	-0.001	**Action rate** $r_{\Delta a} := \lVert \mathbf{a}_t - \mathbf{a}_{t-1} \rVert_2^2$	-0.01
Joint acceleration $r_{\ddot{q}} := \sum_{j \in \mathcal{J}_{\text{lower}}} \ddot{q}_j^2$	-2.5×10^{-7}	**Vertical velocity** $r_{v_z} := v_z^2$	-1.0
Roll/pitch rate $r_{\omega_{xy}} := \omega_x^2 + \omega_y^2$	-0.15		
Contact & Gait			
Undesired contacts $r_{\text{uc}} := \sum_{b \in \mathcal{B}_{\text{nonfoot}}} \mathbb{I}\left(\max_{\tau \in [t-\Delta, t]} \lVert \mathbf{F}_b(\tau) \rVert_2 > F_{\text{thr}}\right)$	-1.0	**Feet slide** $r_{\text{slide}} := \sum_{f \in \{L, R\}} \lVert \mathbf{v}_f^{xy} \rVert_2 \cdot \mathbb{I}(f \text{ in contact})$	-0.25
Flying $r_{\text{fly}} := \mathbb{I}(\text{no foot contact})$	-1.0	**Feet force** $r_F := \sum_{f \in \{L, R\}} \text{clip}\left(\max(\lvert F_{z,f} \rvert - 500, 0), 0, 400\right)$	-0.003
Feet air-time $r_{\text{air}} := \min(t_{\text{air}}, 0.4) \cdot \mathbb{I}(\text{single-stance}) \cdot \mathbb{I}\left(\lVert \mathbf{v}^*_{xy} \rVert_2 + \lvert \omega_z^* \rvert > 0.1\right)$	0.15	**Feet stumble** $r_{\text{stumble}} := \mathbb{I}\left(\exists f \in \{L, R\} : \lVert \mathbf{F}_f^{xy} \rVert_2 > 5 \lvert F_{z,f} \rvert\right)$	-2.0
Stability			
Torso orientation $r_{\text{ori}} := \lVert \mathbf{g}_{\text{torso},xy} \rVert_2^2$	-2.0	**Joint limits** $r_{\text{lim}} := \sum_j \left(\max(q_j - q_j^{\max}, 0) + \max(q_j^{\min} - q_j, 0)\right)$	-2.0
Flat orientation $r_{\text{flat}} := \lVert \mathbf{g}_{xy} \rVert_2^2$	-1.0	**Feet distance** $r_{\text{near}} := \max(d_{\text{thr}} - \lVert \mathbf{p}_L - \mathbf{p}_R \rVert_2, 0)$	-2.0
Other			
Joint deviation $r_{\text{dev}}(\mathcal{J}) := \sum_{j \in \mathcal{J}} \lvert q_j - q_j^{\text{default}} \rvert$	-0.02 to -0.2	**Termination** $r_{\text{term}} := \mathbb{I}(\text{terminated})$	-200

cates stumbling or unstable foot-ground interaction.

d) Stability terms:

- **Torso orientation penalty.** Penalizes deviation of the torso from an upright orientation, helping maintain whole-body balance.
- **Joint limits penalty.** Penalizes violations of soft joint position limits, preventing the policy from relying on unrealistic or unsafe joint configurations.
- **Flat orientation penalty.** Penalizes non-flat base orientation, further encouraging upright and balanced locomotion behavior.
- **Feet distance penalty.** Penalizes configurations in which the two feet become too close to each other, which helps maintain a reasonable support polygon and improves balance robustness.

e) Other terms:

- **Joint deviation penalty.** Penalizes deviation from default joint configurations for selected joint groups, encouraging natural posture priors and preventing excessive joint drift.
- **Termination penalty.** Applies a large penalty when the episode terminates, strongly discouraging catastrophic failure such as falling or unrecoverable instability.