HAF: Adapting Generalist VLAs to Humanoid Whole-Body Loco-manipulation via Hierarchical Action Flow and Spectral Latent RL

Langzhe Gu[1,2*], Chengkai Hou[1,2*], Meng Li[2*], Xinhua Wang[2], Jiaming Liu[1], Xinyuan Lv[2,3], Bowei Zhang[2,3], Shuanghao Bai[2,4], Guangrun Li[1,2], Jingyang He[1,2], Gaole Dai[1], Ziluo Ding[2], Zhiyuan Xu[2], Kuan Cheng[1], Jian Tang[2✉], Zhengping Che[2✉], Shanghang Zhang[1✉]

[1]State Key Laboratory of Multimedia Information Processing,
School of Computer Science, Peking University
[2]Beijing Innovation Center of Humanoid Robotics
[3]Nankai University [4]Xi'an Jiaotong University
[*]Equal contribution. [✉]Corresponding author.

Abstract: Humanoid robots hold great promise as general-purpose agents in human-centered environments, yet generalist vision-language-action (VLA) foundation models are not readily applicable to humanoid whole-body loco-manipulation. The high dimensionality and interdependence of humanoid whole-body motions make it challenging for conventional single-stage VLA architectures to coordinate locomotion, waist posture, and dual-arm manipulation effectively. Moreover, policies trained purely through offline behavior cloning can remain suboptimal during real-world deployment, motivating further policy refinement through real-world interaction. Although online reinforcement learning can refine policies without additional human demonstrations, directly tuning large VLA backbones demands excessive computation and may introduce safety risks during real-robot exploration. To address these bottlenecks, we introduce HAF (Humanoid Adaptation Framework), a two-part framework consisting of HAF-VLA and HAF-Steer that transfers off-the-shelf generalist VLA foundation models to humanoid whole-body loco-manipulation. HAF-VLA is the core hierarchical action-flow generator built on a pretrained flow-matching VLA. It splits full-body action denoising into three sequential stages (locomotion/head $\rightarrow$ + waist $\rightarrow$ + dual-arm manipulation) with stage embeddings and cross-stage KV caches that retain inherent kinematic dependencies, avoiding incoherent whole-body actions from one-shot full-action generation. On top of the frozen HAF-VLA, we design HAF-Steer, a latent offline-to-online RL pipeline. By leveraging the invertibility of flow matching and DCT-based dimensionality reduction, it restricts RL optimization to a compact noise subspace and trains a regularized SAC policy. This avoids updating the large VLA backbone and enables efficient offline-to-online refinement of real-world policy performance. Evaluated on seven real-world humanoid loco-manipulation tasks, HAF surpasses vanilla single-stage VLA baselines and achieves improved whole-body coordination and task performance in real-world environments.

Keywords: Humanoid Robot, Vision-Language-Action (VLA) models, Reinforcement Learning, Imitation Learning

1 Introduction

Humanoid robots hold the promise of becoming universal agents capable of operating in diverse human-centric environments. Unlike fixed-base or wheeled manipulators, humanoids require the synchronized coordination of locomotion, torso posture, and bimanual manipulation. However, this

complexity introduces a critical challenge: unstable base movements often induce erratic upper-body compensatory motions, which severely degrade balance and manipulation accuracy [1, 2, 3, 4, 5, 6, 7, 8].

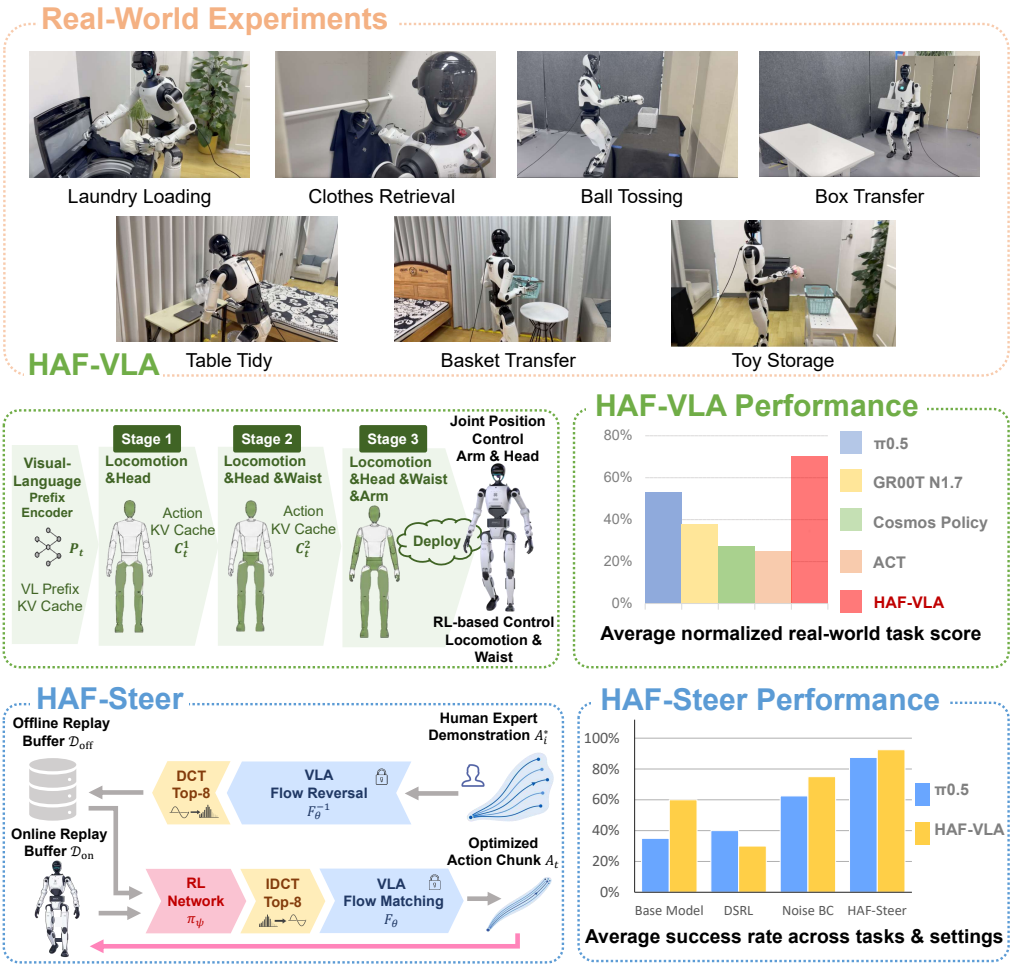


Figure 1: **Overview of HAF.** We demonstrate seven humanoid loco-manipulation tasks in real-world settings. HAF consists of two complementary components: HAF-VLA is a hierarchical action-flow vision-language-action model that generates whole-body motions stage by stage according to the kinematic dependencies among locomotion, torso adjustment, and dual-arm manipulation; HAF-Steer performs latent-space reinforcement learning on discrete cosine transform compressed temporal noise from frozen flow-based VLA policies. Together, they improve task performance in real-world settings.

Although recent generalist vision-language-action (VLA) models have shown strong generalization on conventional robotic platforms [9, 10, 11, 12], adapting them to humanoid whole-body loco-manipulation remains challenging. Standard flow-matching VLAs typically generate all body-part actions in a single stage, without explicitly modeling the dependencies among locomotion, posture, and manipulation [6, 5, 13, 14]. Existing humanoid VLAs often address this issue through humanoid-specific pretraining or large embodiment-specific datasets [8, 11, 15], which substantially increases the cost of adapting a generalist VLA to a new humanoid platform.

Moreover, offline behavior-cloned policies often degrade under deployment-time distribution shifts, yet directly fine-tuning large VLA backbones for high-dimensional humanoid action spaces with online reinforcement learning is computationally expensive and may induce unsafe exploration. Ex-

isting latent-noise RL methods avoid backbone updates but either optimize the full high-dimensional temporal noise or repeat a single noise vector over the entire horizon, sacrificing either efficiency or temporal expressiveness [16, 17, 18, 19].

To address these challenges, we introduce **HAF**, the Humanoid Adaptation Framework for transferring pretrained flow-matching VLAs to humanoid whole-body loco-manipulation. **HAF-VLA** restructures whole-body action generation according to the dependency among locomotion, body posture, and manipulation, while **HAF-Steer** performs lightweight policy refinement in a compact latent noise space. The two components operate at complementary levels: HAF-VLA provides a structured whole-body action generator, while HAF-Steer further refines the frozen generator through real-world offline-to-online adaptation.

HAF-VLA repurposes a pretrained generalist VLA [10] by orchestrating a progressive, kinematics-aware generation process. Instead of producing full-body actions in a single shot, HAF-VLA sequentially outputs commands for locomotion and head orientation, followed by torso adjustment, and finally bimanual manipulation. This hierarchical ordering prioritizes base stabilization to mitigate the erratic compensatory motions often observed in upper-body control [20, 21]. To ensure global coherence, we employ cross-stage KV-cache conditioning: clean actions from earlier stages are re-encoded into cross-stage KV caches to condition subsequent generation. Since the active action sets are cumulative, later stages can refine previously generated dimensions, and only the final full-body action chunk is executed.

Built upon the frozen VLA model, **HAF-Steer** introduces a low-dimensional spectral action space for offline-to-online reinforcement learning (RL). We numerically reverse the frozen flow matching field to recover the initial noise corresponding to demonstrated action chunks and apply the discrete cosine transform (DCT) along the temporal dimension, retaining only the first 8 coefficients. This design avoids the high computational cost of optimizing in the raw high-dimensional action space or fine-tuning the massive VLA backbone, while preserving smoothly varying temporal modes. An RL actor for spectral action is trained through behavior cloning (BC) with offline expert spectral samples first and subsequently optimized using mixed offline–online Soft Actor-Critic (SAC) [22] with BC regularization.

We validate HAF on two physical humanoid robots across seven complex household loco-manipulation tasks. HAF-VLA significantly outperforms imitation learning and standard VLA baselines, particularly in tasks requiring long-distance travel and coordinated full-body motion. With the HAF-Steer module, our framework achieves robust performance gains in both in-distribution and challenging OOD scenarios. Our contributions are summarized as follows:

- **Humanoid Adaptation Framework:** We propose **HAF**, an effective framework that repurposes pretrained generalist VLAs for humanoid whole-body loco-manipulation, eliminating the need to train humanoid-specific foundation models from scratch.
- **Kinematics-Aligned Generation:** We design **HAF-VLA**, a hierarchical action-flow module that aligns with humanoid kinematics to suppress unstable compensatory movements and improve action coherence in whole-body loco-manipulation.
- **Efficient Real-World Refinement:** We develop **HAF-Steer**, a DCT-based latent RL pipeline that enables stable mixed offline–online adaptation for frozen VLA backbones.
- **Comprehensive Real-World Validation:** We validate HAF on two humanoid platforms across seven real-world loco-manipulation tasks, demonstrating improved task performance, whole-body coordination, and robustness under distribution shifts.

2 Related Work

Humanoid VLAs. Vision-language-action models (VLAs) aim to map visual observations and language instructions directly to robot actions [23, 24, 25, 26]. Representative works such as RT-2 [27] and OpenVLA [24] show that vision-language pretraining can be effectively transferred to

robotic manipulation, while π_0 and $\pi_{0.5}$ further move toward continuous generative action modeling with flow matching [9, 10]. However, most existing VLAs still focus on tabletop tasks or mobile manipulators, leaving high-dimensional humanoid whole-body action generation underexplored. Recent works have begun to extend VLAs to humanoid robots [28, 29, 30, 31]. For instance, GR00T N1.7 [11] studies general-purpose humanoid action generation, WholeBodyVLA [8] explores latent VLA learning for large-space mobile manipulation, and Ψ_0 [15] introduces an open foundation model pretrained on egocentric human videos. In contrast to these methods that rely on latent skills, cross-embodiment transfer, or learning from human videos, our method is directly built upon a pretrained generalist VLA. We improve its performance on humanoid robots by generating structured whole-body actions grounded in the forward-kinematic chain, sparing the computational cost of training a new model.

RL post-training for VLA models. Pre-trained VLA policies obtained solely through offline behavior cloning can remain suboptimal during real-world deployment, motivating reinforcement learning (RL) for post-training refinement [32, 33, 34, 35, 36, 18]. Current approaches fall into three categories. The first directly fine-tunes the entire VLA backbone using algorithms like PPO [34, 37, 38], but this incurs massive computational overhead and risks generating unstable, unsafe motions that erase pre-trained priors in high-dimensional whole-body tasks. The second freezes the VLA backbone and trains lightweight residual networks to revise raw actions (e.g., ResFit [39], EXPO [40], DICE-RL [41]). Despite variance reduction techniques, these methods still optimize in the original action space, leading to low sample efficiency and accumulated temporal errors in long-horizon tasks. The third branch performs RL within the latent noise space, which is most relevant to our work. However, existing methods also have some limitations: DSRL [16] reduces the search dimension by repeating a single noise vector across the entire action chunk, but this construction departs largely from the independently sampled Gaussian temporal noise used during VLA pretraining. CFGRL [42] only supports test-time modulation without iterative online optimization, while FRS [17] lacks frequency-domain dimensionality reduction for temporal noise. Compared with prior latent RL work, HAF-Steer significantly reduces the search dimension of temporal noise, suppresses unstable high-frequency exploratory movements during real-world interaction, and enables more efficient real-world adaptation on long-horizon humanoid loco-manipulation tasks.

Whole-body manipulation. Recent research on humanoid whole-body control lays an essential technical foundation for integrated locomotion and manipulation tasks. Advanced low-level controllers and RL-based motion pipelines enable legged robots to perform agile, dynamic movements [43, 44, 6, 45]. Language-conditioned whole-body controllers such as LangWBC [28] and LeVERB [29] support high-level language-guided navigation but lack fine-grained bimanual dexterous manipulation capabilities. VR teleoperation frameworks including TWIST2 [46], AMO [47] and SONIC [43] deliver effective pipelines to collect full-body humanoid trajectories and optimize low-level motion tracking, yet they merely serve as data acquisition tools rather than generalizable vision-language-driven policies for long-horizon loco-manipulation. None of these whole-body control architectures are integrated with hierarchical generative VLA backbones.

3 HAF-VLA: Hierarchical Action Flow for Humanoid Whole-Body Loco-Manipulation

HAF-VLA is a hierarchical action-flow module developed to repurpose pretrained generalist flow-matching VLA foundation models for humanoid whole-body loco-manipulation. Unlike single-stage VLA generators that entangle all body movements, it reflects humanoid kinematic dependencies by producing motions step by step: locomotion and head signals are predicted first, followed by waist adjustment, and finally fine bimanual manipulation trajectories. To link motion outputs across generation phases, we introduce a cross-stage KV-cache conditioning mechanism that feeds encoded action features from earlier stages as context to subsequent generation branches, unifying shared visual-linguistic representations and reducing unstable upper-body compensatory movements triggered by flawed base or torso poses.

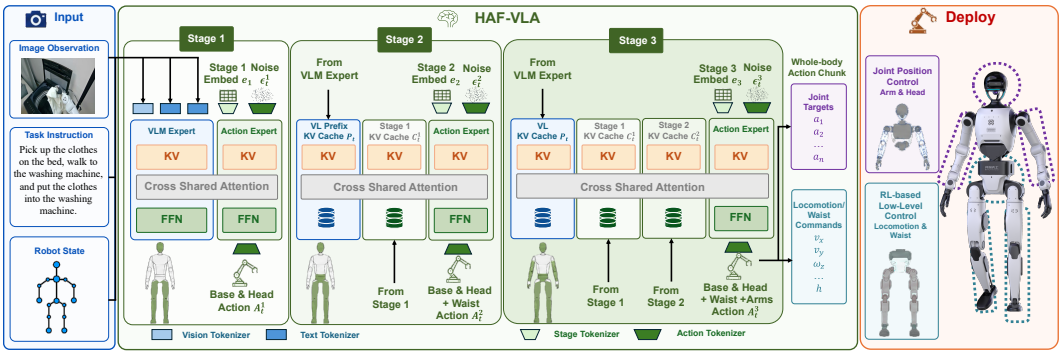


Figure 2: **Overview of HAF-VLA.** A shared action expert progressively expands the active action space from locomotion and head control to waist posture and bimanual manipulation. Clean action caches from earlier stages condition subsequent generation, and only the final full-body action chunk is executed.

3.1 Hierarchical Whole-Body Action Generation

Conditioned on the current observation $o_t = (I_t, q_t)$, which consists of the egocentric RGB image I_t and robot proprioception q_t, as well as the language instruction ℓ, the VLA model predicts a structured whole-body action chunk A_t. Formally, the action chunk is defined as a sequence of future control steps: $A_t = [a_t, \ldots, a_{t+H-1}] \in \mathbb{R}^{H \times D}$, where H denotes the predicted action horizon length and D denotes the dimension of per-step whole-body action. In our implementation, $H = 100$ and the robot executes the first 40 steps before the next inference call. To support hierarchical action generation for whole-body loco-manipulation, each instantaneous action $a_t \in \mathbb{R}^D$ is explicitly decomposed into four orthogonal kinematic sub-components: $a_t = [a_t^{\text{move}}, a_t^{\text{head}}, a_t^{\text{waist}}, a_t^{\text{manip}}]$, which correspond to locomotion and skill-mode commands, head orientation control, waist posture adjustment, and bimanual manipulation targets.

Based on this explicit action decomposition, we further partition the full D-dimensional action index space into four mutually disjoint subsets corresponding to the above kinematic components: $\mathcal{A}_t^{\text{move}}$, $\mathcal{A}_t^{\text{head}}$, $\mathcal{A}_t^{\text{waist}}$, and $\mathcal{A}_t^{\text{manip}}$. Instead of predicting all action dimensions simultaneously in a single forward pass, HAF-VLA achieves progressive, coarse-to-fine whole-body generation via three nested cumulative action index sets constructed from the four subsets:

$$\mathcal{A}_t^1 = \mathcal{A}_t^{\text{move}} \cup \mathcal{A}_t^{\text{head}}, \qquad \mathcal{A}_t^2 = \mathcal{A}_t^1 \cup \mathcal{A}_t^{\text{waist}}, \qquad \mathcal{A}_t^3 = \mathcal{A}_t^2 \cup \mathcal{A}_t^{\text{manip}}. \tag{1}$$

This construction strictly establishes the nested inclusion relation $\mathcal{A}_t^1 \subset \mathcal{A}_t^2 \subset \mathcal{A}_t^3$, supporting hierarchical action generation across three progressive stages. In Stage 1, the model generates fundamental locomotion behaviors, task skill modes, and gaze directions over the minimal action set $\mathcal{A}_t^1$. Stage 2 expands the active action space to $\mathcal{A}_t^2$, incorporating waist actuation signals to refine torso posture and reshape the upper-body workspace. Stage 3 fully activates the complete action space $\mathcal{A}_t^3$ to output the final executable whole-body action commands. Different from rigid decoupled generation schemes that fix early-stage outputs, our cumulative nested design allows each subsequent stage to refine and update action dimensions predicted in earlier stages, enabling more accurate and coherent whole-body loco-manipulation.

All three hierarchical stages share identical weights for the action-flow expert network. To reduce redundant computation, the vision-language prefix KV cache is computed only once at the start and reused across all stages of hierarchical inference: $P_t = F_{\text{prefix}}(o_t, \ell)$, where P_t is the shared vision-language prefix cache derived from the current observation o_t and language instruction ℓ. Each stage starts sampling from an independent Gaussian noise vector $\epsilon_t^s \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and stage-specific behavior is modulated via a trainable stage embedding e_s for stage index $s \in \{1, 2, 3\}$. The full

multi-stage generation process is formalized below:

$$\begin{aligned} A_t^1 &= F_\theta(\epsilon_t^1 \mid P_t, e_1), & C_t^1 &= \mathrm{Cache}_\theta(A_t^1 \mid P_t, e_1), \\ A_t^2 &= F_\theta(\epsilon_t^2 \mid P_t, C_t^1, e_2), & C_t^2 &= \mathrm{Cache}_\theta(A_t^2 \mid P_t, C_t^1, e_2), \\ A_t^3 &= F_\theta(\epsilon_t^3 \mid P_t, C_t^1, C_t^2, e_3). \end{aligned} \tag{2}$$

Here F_θ denotes the shared numerical action-flow map, with the stage-specific action mask applied internally according to the corresponding stage. The $\mathrm{Cache}_\theta(\cdot)$ operator re-encodes the denoised action prediction from the current stage and compresses it into an action KV cache (C_t^1 for Stage 1, C_t^2 for Stage 2). These cached motion embeddings inject coarse prior motion context into later generation steps. Critically, only the final output chunk $A_t \equiv A_t^3$ is sent to the robot controller for real execution; intermediate predictions A_t^1 and A_t^2 serve purely as context providers and are never deployed directly.

3.2 Training Strategy and Deployment

Let $m_s \in \{0,1\}^D$ be the binary indicator of the active action set $\mathcal{A}_t^s$, broadcast along the temporal dimension. For compactness, we denote the conditioning of the three stages as $h_t^1 = (P_t, e_1)$, $h_t^2 = (P_t, C_t^1, e_2)$, and $h_t^3 = (P_t, C_t^1, C_t^2, e_3)$. Given a clean action chunk $A_t \in \mathbb{R}^{H\times D}$, each Stage s independently samples Gaussian noise $\epsilon_t^s \sim \mathcal{N}(0, I)$ and flow time $\tau^s \sim \mathcal{U}(0,1)$. The noisy input and target velocity for Stage s are

$$\begin{aligned} X_{\tau^s,t}^s &= [(1-\tau^s)\epsilon_t^s + \tau^s A_t] \odot m_s, \\ u_t^s &= (A_t - \epsilon_t^s) \odot m_s, \end{aligned} \tag{3}$$

where $X_{\tau^s,t}^s$ is the masked noisy action input and u_t^s is the corresponding masked flow-matching target. All stages operate in the same global action space. The stage mask is applied to both the input and target, such that inactive dimensions are assigned zero target velocity rather than excluded from optimization.

The shared action expert is trained with the stage-wise flow-matching objective

$$\mathcal{L}_{\mathrm{HAF}} = \mathbb{E}_{A_t, \{\epsilon_t^s, \tau^s\}_{s=1}^3} \left[\sum_{s=1}^{3} \frac{1}{HD} \left\| v_\theta\left(X_{\tau^s,t}^s, \tau^s; h_t^s\right) - u_t^s \right\|_F^2 \right], \tag{4}$$

where v_θ denotes the velocity field predicted by the shared action expert. The three stage losses are summed with equal weights. Apart from their independently sampled noise and flow time, the stages differ only in their conditioning h_t^s and active action mask m_s, while sharing the same model parameters.

During training, teacher forcing is used to compute cross-stage caches C_t^1 and C_t^2 from re-encoded masked ground-truth actions rather than stage outputs, which prevents error accumulation while retaining the hierarchical inference structure. At inference, the three stages run sequentially with independent Gaussian noise samples. C_t^1 and C_t^2 are built from the denoised outputs of earlier stages to condition later ones, and only Stage 3's result is executed via receding-horizon control. Each stage starts from independently sampled noise and performs 10 flow-denoising steps conditioned on the visual-language cache and the available cross-stage action caches.

We deploy HAF-VLA with a receding-horizon execution scheme. The robot executes the first 40 steps before the next inference call. In our real-robot deployment, the complete three-stage inference takes approximately 0.12 seconds on a single RTX 5090 GPU, supporting real-time closed-loop humanoid loco-manipulation.

4 HAF-Steer: Spectral Latent RL for Flow-Matching VLAs

HAF-Steer is a lightweight reinforcement-learning method for adapting a frozen flow-matching VLA through its initial flow noise, as illustrated in Figure 3. Demonstrated actions are first mapped back to their corresponding noise and compressed into low-dimensional spectral actions. A stochastic

policy is then trained in this spectral space and decoded through the frozen flow generator to produce executable action chunks. We first define the spectral latent space and then introduce its offline-to-online optimization.

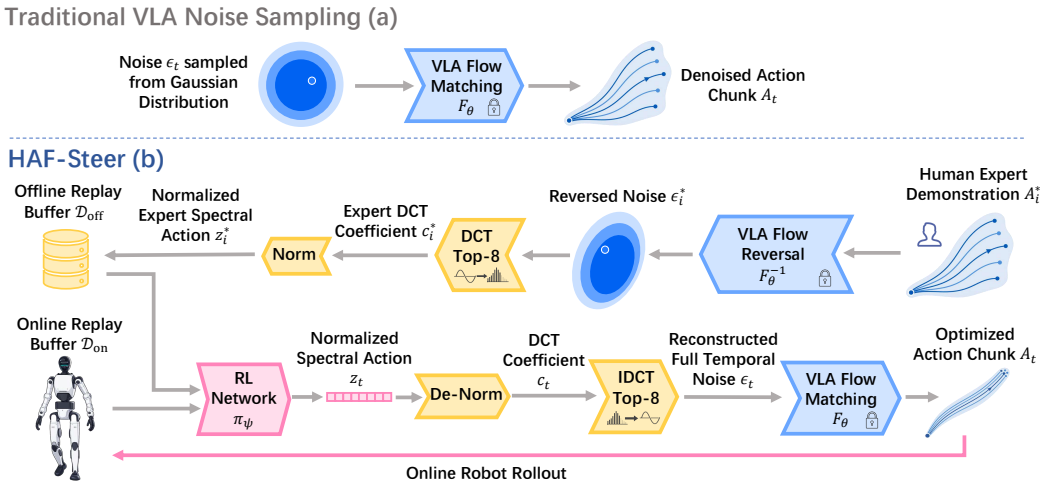


Figure 3: **Overview of HAF-Steer. (a)** A conventional flow-matching VLA samples full-dimensional Gaussian noise and decodes it into an action trajectory through a frozen flow generator. **(b)** HAF-Steer instead constructs a low-dimensional spectral action space from expert demonstrations: demonstrated trajectories are mapped back to their initial flow noise through numerical reverse integration, compressed by retaining the first 8 temporal DCT coefficients, and normalized to form expert spectral actions in the offline replay buffer. An RL policy is trained with mixed offline and online experience to predict spectral actions, which are de-normalized, reconstructed into full temporal noise by inverse DCT, and decoded by the frozen VLA to produce executable robot trajectories.

4.1 Spectral Latent Space via Flow Reversal

HAF-Steer builds upon a frozen pre-trained flow-matching VLA backbone F_θ, which generates whole-body action chunks via a deterministic flow mapping: $A_t = F_\theta(\epsilon_t \mid \zeta_t), \quad \epsilon_t \in \mathbb{R}^{H\times D}$, where H is the action chunk length, D is the dimension of robot action, $A_t \in \mathbb{R}^{H\times D}$ denotes the predicted action chunk, ϵ_t is the initial flow noise, and ζ_t represents the unified vision-language-robot conditioning of the frozen VLA model. The network parameters θ are fully fixed during subsequent reinforcement learning adaptation.

Directly optimizing raw temporal noise at all H time steps leads to an excessively high-dimensional exploration space. Prior steering methods [16, 17] repeat a single noise vector over the entire chunk, which we found unstable and frequently inducing severe body jerking or unsafe transient behaviors in whole-body RL training. To enforce temporal smoothness and compress action space complexity, we constrain RL exploration within a low-dimensional spectral subspace spanned by the first K discrete cosine transform (DCT) bases. In our practice, we empirically choose $K = 8$, which greatly reduces the exploration space from $H \times D$ to $K \times D$. Our multi-mode spectral parameterization retains diverse smooth temporal variation patterns and yields physically plausible whole-body action corrections.

Based on the above spectral dimensionality reduction design, we further build offline supervised spectral targets to guide policy training. Specifically, to construct supervised spectral targets from offline demonstrations, we first invert each expert action chunk back to its corresponding flow noise space using the frozen VLA model. Given a demonstrated action A_i^* and its conditioning ζ_i^*, we recover and normalize its spectral representation via DCT transformation:

$$\epsilon_i^* = F_\theta^{-1}(A_i^* \mid \zeta_i^*), \quad c_i^* = \mathrm{DCT}_K(\epsilon_i^*), \quad z_i^* = \frac{c_i^* - \mu_c}{\sigma_c + \delta}. \tag{5}$$

Here, F_θ^{-1} denotes numerical backward integration of the fixed flow field rather than a learned inverse model. ϵ_i^* is the recovered full-scale flow noise, $c_i^* \in \mathbb{R}^{K\times D}$ truncates the first K DCT temporal coefficients, and z_i^* is the normalized expert spectral action. The statistics μ_c, σ_c are precomputed over the entire demonstration set, and δ is a small constant for numerical stability. According to the derived spectral expert representations, we construct an offline spectral replay buffer for policy training. We formulate VLA-level transition tuples consisting of spectral expert targets, sparse task rewards, and episodic termination flags:

$$\mathcal{D}_{\text{off}} = \{(x_i, z_i^*, r_i, x_{i+1}, d_i)\}_{i=1}^{N-1}, \tag{6}$$

Here, x_i and x_{i+1} denote consecutive observational states, z_i^* is the precomputed spectral expert target, and d_i indicates episode termination. We adopt a sparse reward scheme for demonstration-guided training: terminal transitions of successful trajectories are assigned $r_i = 1$, while all intermediate transitions receive $r_i = 0$. This simple yet effective reward formulation drives the policy to learn task-aligned spectral noise corrections from offline data.

4.2 Offline-to-Online Spectral Policy Learning

Given the spectral action space and offline replay buffer constructed above, we first train the actor through behavior cloning and then improve it using mixed offline–online reinforcement learning. After training, the learned spectral actor generates smooth whole-body corrections at inference time through DCT spectral decoding.

Behavior-cloning initialization. Let $\mu_\psi(x)$ denote the mean output of the stochastic spectral actor $\pi_\psi(z \mid x)$. We first initialize the actor by regressing toward the recovered expert spectral actions in $\mathcal{D}_{\text{off}}$:

$$\mathcal{L}_{\text{BC}} = \mathbb{E}_{(x_i, z_i^*)\sim\mathcal{D}_{\text{off}}}\left[\|\mu_\psi(x_i) - z_i^*\|_2^2\right]. \tag{7}$$

Only the spectral actor is optimized during this initialization stage; the VLA backbone and value functions remain unchanged.

Mixed Offline–Online Reinforcement Learning. After BC initialization, the policy interacts with the real robot following the spectral generation pipeline in Eq. (9), and newly collected transitions are stored in an online replay buffer $\mathcal{D}_{\text{on}}$. We adopt the same sparse terminal reward rule used for offline data: only successful terminal steps receive $r = 1$, while all other transitions yield $r = 0$. In each training iteration, we sample mixed mini-batches $\mathcal{B} = \mathcal{B}_{\text{off}} \cup \mathcal{B}_{\text{on}}$ from both buffers. All samples participate in standard SAC policy updates, while BC regularization is only applied to offline expert data to preserve demonstration quality:

$$\begin{aligned}\mathcal{L}_\pi = \mathbb{E}_{x_i\sim\mathcal{B},\, z_i\sim\pi_\psi(\cdot|x_i)}&\left[\alpha \log \pi_\psi(z_i \mid x_i) - \min_j Q_{\omega_j}(x_i, z_i)\right] \\ &+ \lambda_{\text{BC}}\mathbb{E}_{(x_i, z_i^*)\sim\mathcal{B}_{\text{off}}}\left[\|\mu_\psi(x_i) - z_i^*\|_2^2\right].\end{aligned} \tag{8}$$

Here, α denotes the SAC entropy temperature, $\{Q_{\omega_j}\}$ are twin critic networks, and λ_{BC} balances demonstration regularization strength. The critics are updated via standard entropy-regularized SAC Bellman loss using the full mixed mini-batch. This hybrid training scheme enables offline demonstrations to stabilize early policy learning and online interactions to adapt to real-world deployment distributions. We gradually decay the offline sampling ratio throughout training to shift optimization from demonstration behavior toward real robot experience. Consistent with our adaptation protocol, the entire flow-matching VLA backbone remains frozen; only the spectral actor, critics, target critics, and entropy temperature are updated during training.

Inference Pipeline. At test time, the trained spectral actor π_ψ operates purely in the normalized spectral space. Given the current observation x_t, the model samples spectral actions, recovers full temporal noise via inverse DCT, and generates refined whole-body action chunks for robot execution:

$$z_t \sim \pi_\psi(\cdot \mid x_t), \quad c_t = \mu_c + (\sigma_c + \delta) \odot z_t, \quad \epsilon_t = \text{IDCT}_K(c_t), \quad A_t = F_\theta(\epsilon_t \mid \zeta_t) \tag{9}$$

The recovered noise ϵ_t is fed into the frozen VLA flow model conditioned on ζ_t to produce final executable whole-body actions. The IDCT_K operation zero-pads truncated spectral coefficients to reconstruct smooth full-horizon temporal noise. By restricting RL optimization to the low-dimensional spectral subspace ($K \ll H$), our framework significantly reduces optimization complexity, suppresses high-frequency jitter, and yields robust, physically plausible whole-body loco-manipulation behaviors.

4.3 Instantiation on Flow-Matching VLA Backbones

The formulation above is independent of a particular flow-matching VLA architecture. We apply the same spectral parameterization, flow-reversal procedure, and offline-to-online objective to both $\pi_{0.5}$ and HAF-VLA.

For $\pi_{0.5}$, F_θ is its native action-flow map and ζ_t denotes its standard visual–language and robot-state conditioning. No change to the pretrained VLA parameters is required.

For HAF-VLA, we instantiate the generic flow map only on its final generation stage. Stages 1 and 2 execute normally and provide their clean-action caches, while HAF-Steer replaces only the initial noise of Stage 3. During demonstration inversion, C_t^1 and C_t^2 are constructed through teacher forcing; during deployment, they are obtained from the predicted clean actions of the first two stages. In both $\pi_{0.5}$ and HAF-VLA, the complete VLA backbone remains frozen.

5 Experiments

In this section, we aim to answer four key research questions: **Q1.** Does HAF-VLA enable long-horizon humanoid loco–manipulation beyond existing state-of-the-art methods? **Q2.** Does HAF-VLA's hierarchical action-flow design improve imitation-learning performance? **Q3.** Does HAF-VLA generalize to unseen visual and positional disturbances? **Q4.** Can HAF-Steer improve real-world deployment performance through offline-to-online adaptation?

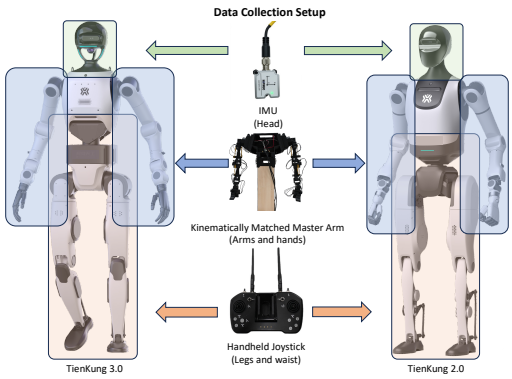


Figure 4: **Robot teleoperation data collection setup.** The teleoperation pipeline uses an IMU for head control, kinematically matched master arms for dual-arm manipulation, and a joystick to command leg locomotion and waist posture.

5.1 Experiment Setup

Hardware and Data Collection. We conduct real-robot experiments on the TienKung 2.0 and TienKung 3.0 humanoid platforms. Perception relies on an onboard egocentric RGB camera mounted on the robot head. Training demonstrations are collected via isomorphic teleoperation: a kinematic master arm controls bimanual manipulation, a handheld joystick governs locomotion and waist motion, and an inertial measurement unit (IMU) tracks head orientation commands, as visualized in Figure 4. We collect 120 teleoperated trajectories for each household task.

Tasks and Evaluation Metric. As illustrated in Figure 5, we benchmark our framework on seven real-world household loco-manipulation tasks: *Laundry Loading*, *Clothes Retrieval*, *Table Tidy*, *Basket Transfer*, *Toy Storage*, *Ball Tossing*, and *Box Transfer*. These tasks demand long-horizon coordination across locomotion, whole-body posture adjustment, dual-arm manipulation, and object interaction, including challenging skills such as traversing separated regions, torso bending, squatting, object carrying, and throwing. Since binary pass/fail success rates cannot capture partial progress during long sequential executions, we adopt a normalized task score as the primary evaluation metric. Each task is split into predefined milestone sub-goals with individual scores. For each trial rollout,

we compute the normalized score as:

$$\text{Score}(\%) = \frac{s_{\text{achieved}}}{s_{\max}} \times 100.$$

We report the average normalized score across 10 independent rollouts per method and task, which enables fine-grained quantitative comparison across tasks with different completion criteria.

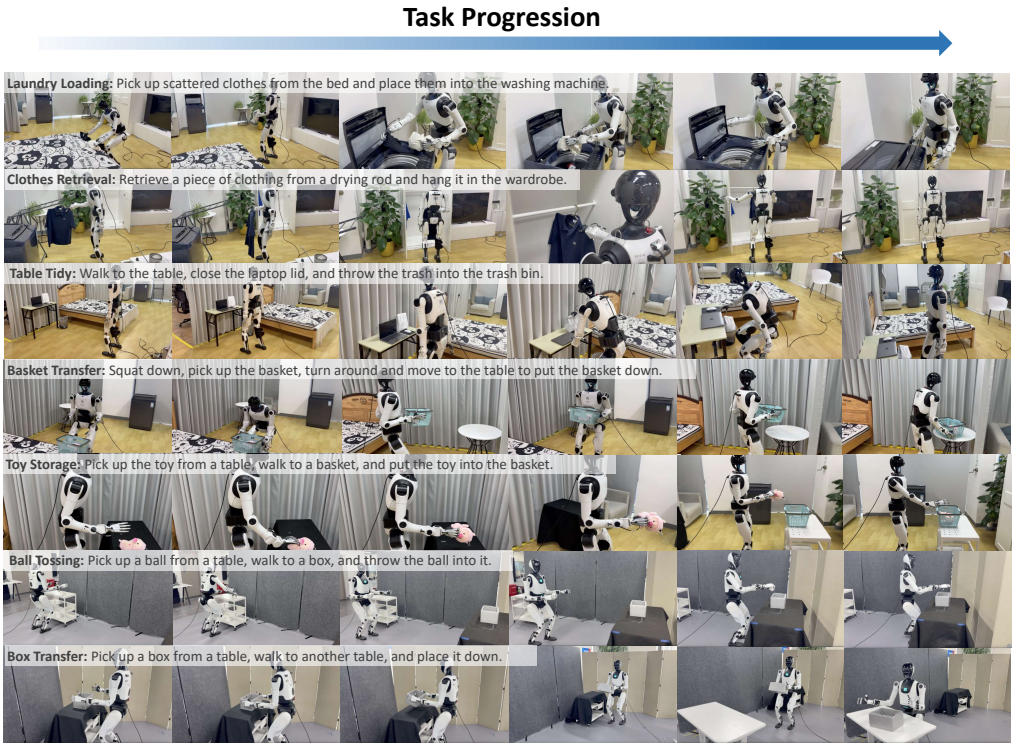


Figure 5: **Seven real-world humanoid loco-manipulation tasks.** Each row depicts a representative execution sequence requiring navigation, whole-body posture adjustment, and physical object interaction.

Baselines. We compare our approach against four representative state-of-the-art baselines: (i) ACT [48], a well-established action-chunking imitation learning algorithm; (ii) $\pi_{0.5}$ [10], a strong generalist flow-matching vision-language-action policy; (iii) GR00T N1.7 [11], a large-scale pre-trained humanoid foundation model for general robot skill learning; and (iv) Cosmos Policy [12], a recent world-model-based framework for robotic manipulation. We strictly follow the official open-source implementations and recommended hyperparameters for all baselines.

5.2 Q1: Does HAF-VLA enable long-horizon humanoid loco–manipulation beyond existing SOTA methods?

We evaluate HAF-VLA across seven long-horizon humanoid loco-manipulation tasks requiring navigation, whole-body coordination, and precise object interaction. As summarized in Table 1, HAF-VLA achieves the best or tied-best average normalized score on all seven tasks, improving the overall average performance from 53.3% with the strongest baseline $\pi_{0.5}$ to 70.5%. Performance gains are particularly pronounced for tasks combining locomotion with subsequent fine manipulation, including *Box Transfer*, *Ball Tossing*, and *Laundry Loading*. Empirically, both $\pi_{0.5}$ and GR00T N1.7 frequently exhibit locomotion drift, while Cosmos Policy is hindered by relatively high online inference latency. Overall, these results demonstrate the effectiveness of hierarchical action generation for high-dimensional humanoid action spaces in whole-body loco-manipulation.

Table 1: **Main HAF-VLA results on seven long-horizon humanoid loco–manipulation tasks.** We report the average normalized task score (%) over 10 rollout trials. Higher values indicate better performance.

Task	HAF-VLA	$\pi_{0.5}$	GR00T N1.7	Cosmos Policy	ACT
Laundry Loading	**66.7**	53.3	40.0	0.0	10.0
Clothes Retrieval	**53.3**	**53.3**	33.3	26.7	23.3
Table Tidy	**80.0**	70.0	40.0	16.7	23.3
Basket Transfer	**63.3**	50.0	43.3	33.3	16.7
Toy Storage	**80.0**	53.3	30.0	40.0	23.3
Ball Tossing	**56.7**	33.3	36.7	3.3	30.0
Box Transfer	**93.3**	60.0	43.3	73.3	50.0
Average	**70.5**	53.3	38.1	27.6	25.2

5.3 Q2: Does HAF-VLA's hierarchical action-flow design improve imitation-learning performance?

We conduct targeted ablation experiments on the representative *Laundry Loading* task to isolate the contribution of HAF-VLA's hierarchical action-flow design. Full HAF-VLA progressively expands the active action space across three stages, with 10 flow-matching denoising steps per stage and 30 steps in total. We compare it against three variants: (1) *All-Joint Denoising*, where every stage denoises all action dimensions simultaneously; (2) *Arm-First Hierarchy*, which reverses the coarse-to-fine generation order by prioritizing manipulation before locomotion; and (3) vanilla $\pi_{0.5}$ with 30 denoising steps, which controls for the total denoising budget.

Table 2: **Ablation study of HAF-VLA's hierarchical action-flow design.** We report the average normalized task score (%) over 10 rollout trials on the Laundry Loading task. Higher scores are better.

Method / Variant	Stage Design	Total Steps	Norm. Score (%)
HAF-VLA	Locomotion/Head $\rightarrow$ + Waist $\rightarrow$ + Manip.	30	**66.7**
All-Joint Denoising	All joints at every stage	30	53.3
Arm-First Hierarchy	Manip. $\rightarrow$ + Waist $\rightarrow$ + Locomotion/Head	30	50.0
$\pi_{0.5}$, 30 steps	Full-body action	30	20.0

As shown in Table 2, our full hierarchical design outperforms all ablated variants, indicating that the improvement does not simply result from increasing the number of denoising iterations. The inferior performance of the arm-first variant further highlights the importance of establishing locomotion and body posture before fine manipulation. We also observe that increasing $\pi_{0.5}$ from 10 to 30 denoising steps decreases performance despite the larger computation budget, which is mainly caused by the increased inference latency (from 0.075s to 0.115s). Since $\pi_{0.5}$ already exhibits locomotion-prediction errors on humanoid tasks, the additional latency can amplify the temporal mismatch between predicted velocity commands and robot execution. In contrast, HAF-VLA remains effective at a similar latency, suggesting greater robustness to inference delay.

5.4 Q3: Does HAF-VLA generalize to unseen visual and positional disturbances?

We evaluate out-of-distribution robustness under two controlled distribution shifts visualized in Figure 6. For *Laundry Loading*, we place an unseen black office chair along the robot navigation path to introduce visual distraction. For *Clothes Retrieval*, we shift the robot initial position 20 cm backward relative to the demonstration collection setup. We compare HAF-VLA against vanilla $\pi_{0.5}$ using the same normalized task score metric across 10 rollouts per condition. As reported in Table 3, HAF-VLA maintains higher task performance under both visual distraction and positional perturbation, which demonstrates stronger generalization beyond the exact demonstration layout.

Figure 6: **Generalization experiments.** Left: object-distraction test during *Laundry Loading*. Right: position-disturbance test during *Clothes Retrieval*.

Table 3: **Generalization evaluation under controlled disturbances.** We report the average normalized task score (%) over 10 rollout trials. Higher scores indicate stronger robustness.

Task	Disturbance	HAF-VLA	$\pi_{0.5}$
Laundry Loading	Unseen chair near the path	**40.0**	26.7
Clothes Retrieval	20 cm backward start shift	**43.3**	36.7

5.5 Q4: Can HAF-Steer improve real-world deployment performance through offline-to-online adaptation?

We evaluate HAF-Steer on two representative whole-body loco-manipulation tasks, *Toy Storage* and *Basket Transfer*, using the TienKung 3.0 humanoid platform. For each task, we consider both in-distribution (ID) settings covered by the offline demonstrations and out-of-distribution (OOD) settings with unseen goal locations. Specifically, for *Toy Storage*, the destination table is moved 30 cm beyond the spatial range covered during demonstration collection; for *Basket Transfer*, the destination table is similarly shifted by 30 cm from its demonstrated position. These controlled spatial perturbations require the policy to adapt its locomotion and whole-body manipulation behavior to previously unseen goal configurations. To examine whether the proposed spectral adaptation mechanism generalizes across different flow-matching policies, we conduct all experiments on both the vanilla $\pi_{0.5}$ backbone and HAF-VLA. The corresponding ID and OOD evaluation setups are visualized in Figure 7.

Baselines and Variants. We compare four policy variants. *Base Model* directly deploys the frozen VLA without latent adaptation. *DSRL* [16] optimizes a single 32-dimensional noise vector and repeats it across the entire action horizon, thereby reducing the search dimension at the cost of removing temporal variation in the flow noise. *Noise BC* evaluates our spectral actor immediately after the spectral behavior-cloning initialization, without subsequent SAC training. Finally, *HAF-Steer* further optimizes the BC-initialized spectral actor using mixed offline–online SAC while keeping the entire VLA backbone frozen. This comparison isolates both the effect of the spectral noise representation and the additional benefit of online reinforcement learning.

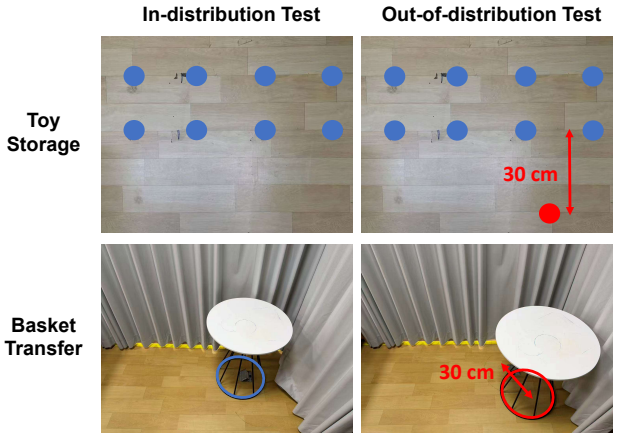
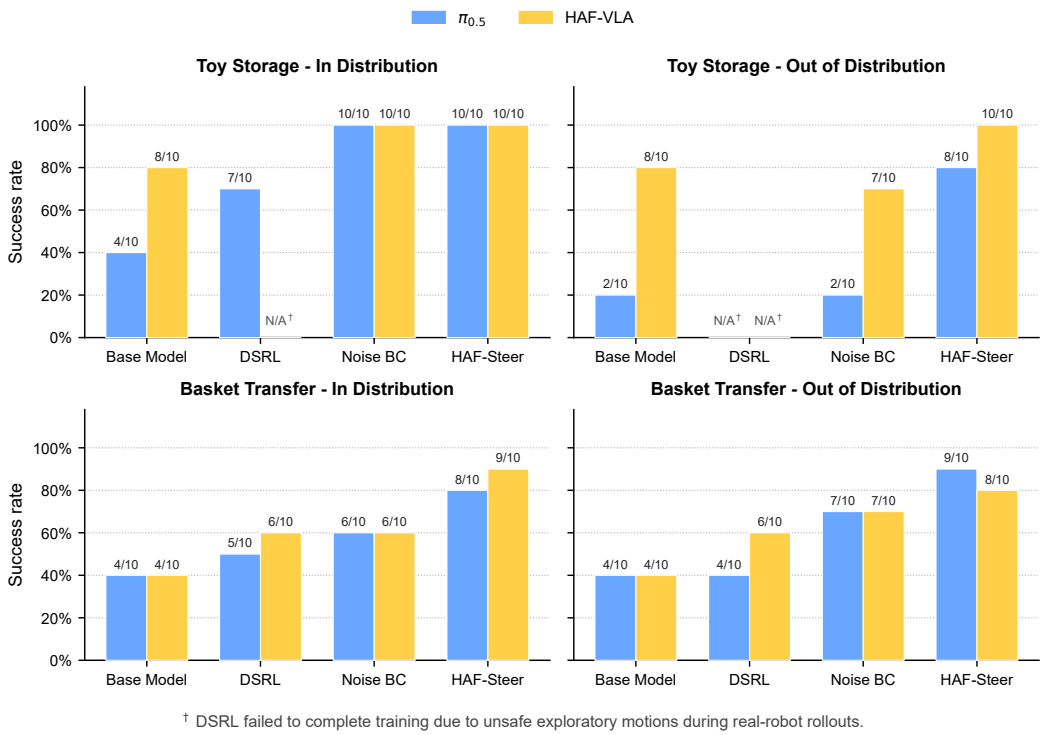


Figure 7: **In-distribution and out-of-distribution evaluation setups for HAF-Steer.** Blue markers indicate destination-table locations covered during demonstration collection, while red markers indicate the OOD evaluation locations. For *Toy Storage*, the destination table is placed 30 cm beyond the demonstrated spatial range. For *Basket Transfer*, the destination table is shifted by 30 cm from its demonstrated position.

Results Analysis. As shown in Figure 8, HAF-Steer consistently improves both $\pi_{0.5}$ and HAF-VLA across ID and OOD settings. Noise

BC already improves performance in several settings, while subsequent mixed offline–online RL further increases real-world success rates, demonstrating that HAF-Steer can refine deployment performance beyond the offline-initialized policy in both ID and OOD conditions. In contrast, the repeated-noise parameterization of DSRL removes temporal variation and produces unsafe exploratory motions in several real-robot experiments, leading to early training termination. By retaining multiple low-frequency temporal modes, HAF-Steer enables more stable latent-space exploration and effective online adaptation.

Importantly, HAF-VLA and HAF-Steer operate at complementary levels. HAF-VLA first provides a structured whole-body policy through hierarchical action generation, while HAF-Steer further adapts this frozen policy to deployment-time distribution shifts. Applying HAF-Steer to HAF-VLA improves its success rate in all four evaluated ID/OOD settings and achieves the best or tied-best performance in three of them. These results demonstrate the benefit of combining structured action generation and lightweight latent policy adaptation within the unified HAF framework.



† DSRL failed to complete training due to unsafe exploratory motions during real-robot rollouts.

Figure 8: **HAF-Steer performance under in-distribution and out-of-distribution settings.** We report successful trials out of 10 rollouts on *Toy Storage* and *Basket Transfer* using both $\pi_{0.5}$ and HAF-VLA backbones. Noise BC denotes the spectral actor after behavior-cloning initialization without SAC post-training. † DSRL failed to complete training because unsafe exploratory motions triggered early termination during real-robot rollouts.

6 Conclusion and Limitations

In this paper, we introduce HAF, a unified framework for adapting pretrained VLA models to humanoid whole-body loco-manipulation. It comprises two key components: HAF-VLA generates structured sequential actions using progressive denoising and cross-stage KV-cache conditioning, while HAF-Steer leverages SAC-based latent reinforcement learning to optimize DCT-compressed noise coefficients and improve real-world deployment performance through offline-to-online adaptation. HAF successfully unifies structured generative modeling and adaptive policy optimization for complex humanoid whole-body loco-manipulation.

Limitations. The hierarchical pipeline increases denoising computation and introduces deployment latency. The latent RL module is also constrained by the base VLA's inherent priors and may therefore fail to correct erroneous motions in extreme unseen scenarios. Future work will explore efficient denoising schemes and lightweight RL implementations to enhance real-time performance and task robustness.

7 Acknowledge

This work was supported by the National Natural Science Foundation of China (62476011), the Beijing Natural Science Foundation (L252060), and the Beijing Major Science and Technology Project under Contract no. Z191100010618003.

References

[1] Q. Ben, F. Jia, J. Zeng, J. Dong, D. Lin, and J. Pang. Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit, 2025. URL https://arxiv.org/abs/2502.13013.

[2] Y. Zhang, Y. Yuan, P. Gurunath, T. He, S. Omidshafiei, A.-a. Agha-mohammadi, M. Vazquez-Chanlatte, L. Pedersen, and G. Shi. FALCON: Learning force-adaptive humanoid loco-manipulation. *arXiv preprint arXiv:2505.06776*, 2025.

[3] Y. Li, Y. Zhang, W. Xiao, C. Pan, H. Weng, G. He, T. He, and G. Shi. Learning gentle humanoid locomotion and end-effector stabilization control. *arXiv preprint arXiv:2505.24198*, 2025.

[4] J. Shi, X. Liu, D. Wang, O. Lu, S. Schwertfeger, F. Sun, C. Bai, and X. Li. Adversarial locomotion and motion imitation for humanoid policy learning. *arXiv preprint arXiv:2504.14305*, 2025.

[5] M. Ji, X. Peng, F. Liu, J. Li, G. Yang, X. Cheng, and X. Wang. ExBody2: Advanced expressive humanoid whole-body control. *arXiv preprint arXiv:2412.13196*, 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. *arXiv preprint arXiv:2508.08241*, 2025.

[7] Y. Ze, Z. Chen, W. Wang, T. Chen, X. He, Y. Yuan, X. B. Peng, and J. Wu. Generalizable humanoid manipulation with 3d diffusion policies. In *IROS*, 2025.

[8] 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 *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=OCJmVjyzN7.

[9] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[10] Physical Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[11] NVIDIA, J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[12] M. J. Kim, Y. Gao, T.-Y. Lin, Y.-C. Lin, Y. Ge, G. Lam, P. Liang, S. Song, M.-Y. Liu, C. Finn, and J. Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=wPEIStHxYH.

[13] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. M. Kitani, C. Liu, and G. Shi. OmniH2O: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. In *Conference on Robot Learning*, 2025.

[14] Z. Zhang, C. Chen, H. Xue, J. Wang, S. Liang, Y. Liu, Z. Zhang, H. Wang, and L. Yi. Unleashing humanoid reaching potential via real-world-ready skill space. *arXiv preprint arXiv:2505.10918*, 2025.

[15] S. Wei, H. Jing, B. Li, Z. Zhao, J. Mao, Z. Ni, S. He, J. Liu, X. Liu, K. Kang, S. Zang, W. Yuan, M. Pavone, D. Huang, and Y. Wang. ψ_0: An open foundation model towards universal humanoid loco-manipulation, 2026. URL https://arxiv.org/abs/2603.12263.

[16] A. Wagenmaker, M. Nakamoto, Y. Zhang, S. Park, W. Yagoub, A. Nagabandi, A. Gupta, and S. Levine. Steering your diffusion policy with latent space reinforcement learning. *arXiv preprint arXiv:2506.15799*, 2025.

[17] A. Tang, W. Chen, A. Wagenmaker, C. Finn, and S. Levine. Improving robotic generalist policies via flow reversal steering. *arXiv preprint arXiv:2606.13675*, 2026.

[18] Y. Li, X. Ma, J. Xu, Y. Cui, Z. Cui, Z. Han, L. Huang, T. Kong, Y. Liu, H. Niu, W. Peng, J. Qiao, Z. Ren, H. Shi, Z. Su, J. Tian, Y. Xiao, S. Zhang, L. Zheng, H. Li, and Y. Wu. Gr-rl: Going dexterous and precise for long-horizon robotic manipulation, 2025. URL https://arxiv.org/abs/2512.01801.

[19] J. Lu, X. Qin, Y. Jiang, K. Wang, C. Zhang, B. Liang, J. Yang, M. Xu, and L. Zhao. Unisteer: Unified noise steering for efficient human-guided vla adaptation, 2026. URL https://arxiv.org/abs/2605.10821.

[20] S. Chen, J. Liu, S. Qian, H. Jiang, L. Li, R. Zhang, Z. Liu, C. Gu, C. Hou, P. Wang, Z. Wang, and S. Zhang. Ac-dit: Adaptive coordination diffusion transformer for mobile manipulation, 2025. URL https://arxiv.org/abs/2507.01961.

[21] Y. Jiang, R. Zhang, J. Wong, C. Wang, Y. Ze, H. Yin, C. Gokmen, S. Song, J. Wu, and L. Fei-Fei. Behavior robot suite: Streamlining real-world whole-body manipulation for everyday household activities, 2025. URL https://arxiv.org/abs/2503.05652.

[22] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018. URL https://arxiv.org/abs/1801.01290.

[23] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation, 2025. URL https://arxiv.org/abs/2410.07864.

[24] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. OpenVLA: An open-source vision-language-action model. In P. Agrawal, O. Kroemer, and W. Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

[25] J. Zheng, J. Li, Z. Wang, D. Liu, X. Kang, Y. Feng, Y. Zheng, J. Zou, Y. Chen, J. Zeng, T. Wang, Y.-Q. Zhang, J. Liu, and X. Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kt51kZH4aG.

[26] Y. Wang, X. Li, W. Wang, J. Zhang, Y. Li, Y. Chen, X. Wang, and Z. Zhang. Unified vision-language-action model, 2025. URL https://arxiv.org/abs/2506.19850.

[27] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, Q. Vuong, V. Vanhoucke, H. Tran, R. Soricut, A. Singh, J. Singh, P. Sermanet, P. R. Sanketi, G. Salazar, M. S. Ryoo, K. Reymann, K. Rao, K. Pertsch, I. Mordatch, H. Michalewski, Y. Lu, S. Levine, L. Lee, T.-W. E. Lee, I. Leal, Y. Kuang, D. Kalashnikov, R. Julian, N. J. Joshi, A. Irpan, B. Ichter, J. Hsu, A. Herzog, K. Hausman, K. Gopalakrishnan, C. Fu, P. Florence, C. Finn, K. A. Dubey, D. Driess, T. Ding, K. M. Choromanski, X. Chen, Y. Chebotar, J. Carbajal, N. Brown, A. Brohan, M. G. Arenas, and K. Han. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In J. Tan, M. Toussaint, and K. Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

[28] Y. Shao, X. Huang, B. Zhang, Q. Liao, Y. Gao, Y. Chi, Z. Li, S. Shao, and K. Sreenath. Langwbc: Language-directed humanoid whole-body control via end-to-end learning, 2025. URL https://arxiv.org/abs/2504.21738.

[29] H. Xue, X. Huang, D. Niu, Q. Liao, T. Kragerud, J. T. Gravdahl, X. B. Peng, G. Shi, T. Darrell, K. Sreenath, and S. S. Sastry. Leverb: Humanoid whole-body control with latent vision-language instruction. *CoRR*, abs/2506.13751, 2025. doi:10.48550/ARXIV.2506.13751. URL https://doi.org/10.48550/arXiv.2506.13751.

[30] P. Ding, J. Ma, X. Tong, B. Zou, X. Luo, Y. Fan, T. Wang, H. Lu, P. Mo, J. Liu, et al. Humanoid-VLA: Towards universal humanoid control with visual integration. *arXiv preprint arXiv:2502.14795*, 2025.

[31] S. Bai, M. Li, X. Lv, J. Wang, X. Wang, F. Liao, C. Hou, L. Gu, W. Zhou, K. Wu, Z. Ding, Z. Xu, L. Sun, S. Zhang, Z. Che, J. Tang, and B. Chen. Hex: Humanoid-aligned experts for cross-embodiment whole-body manipulation, 2026. URL https://arxiv.org/abs/2604.07993.

[32] P. J. Ball, L. Smith, I. Kostrikov, and S. Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pages 1577–1594. PMLR, 2023.

[33] Y. Zhang, L. Ke, A. Deshpande, A. Gupta, and S. Srinivasa. Cherry-picking with reinforcement learning: Robust dynamic grasping in unstable conditions. *arXiv preprint arXiv:2303.05508*, 2023.

[34] K. Lei, H. Li, D. Yu, Z. Wei, L. Guo, Z. Jiang, Z. Wang, S. Liang, and H. Xu. Rl-100: Performant robotic manipulation with real-world reinforcement learning. *arXiv preprint arXiv:2510.14830*, 2025.

[35] J. Luo, Z. Hu, C. Xu, Y. L. Tan, J. Berg, A. Sharma, S. Schaal, C. Finn, A. Gupta, and S. Levine. Serl: A software suite for sample-efficient robotic reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16961–16969. IEEE, 2024.

[36] J. Yang, M. S. Mark, B. Vu, A. Sharma, J. Bohg, and C. Finn. Robot fine-tuning made easy: Pre-training rewards and policies for autonomous real-world reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4804–4811. IEEE, 2024.

[37] D. McAllister, S. Ge, B. Yi, C. M. Kim, E. Weber, H. Choi, H. Feng, and A. Kanazawa. Flow matching policy gradients. *arXiv preprint arXiv:2507.21053*, 2025.

[38] A. Ren, J. Lidard, L. Ankile, A. Simeonov, P. Agrawal, A. Majumdar, B. Burchfiel, H. Dai, and M. Simchowitz. Diffusion policy policy optimization. In *International Conference on Learning Representations*, volume 2025, pages 77288–77329, 2025.

[39] L. Ankile, Z. Jiang, R. Duan, G. Shi, P. Abbeel, and A. Nagabandi. Residual off-policy rl for finetuning behavior cloning policies. *arXiv preprint arXiv:2509.19301*, 2025.

[40] P. Dong, Q. Li, D. Sadigh, and C. Finn. Expo: Stable reinforcement learning with expressive policies. *arXiv preprint arXiv:2507.07986*, 2025.

[41] Z. Sun and S. Song. From prior to pro: Efficient skill mastery via distribution contractive rl finetuning. *arXiv preprint arXiv:2603.10263*, 2026.

[42] K. Frans, S. Park, P. Abbeel, and S. Levine. Diffusion guidance is a controllable policy improvement operator. *arXiv preprint arXiv:2505.23458*, 2025.

[43] 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, 2026. URL https://arxiv.org/abs/2511.07820.

[44] Y. Li, Z. Luo, T. Zhang, C. Dai, A. Kanervisto, A. Tirinzoni, H. Weng, K. Kitani, M. Guzek, A. Touati, A. Lazaric, M. Pirotta, and G. Shi. Bfm-zero: A promptable behavioral foundation model for humanoid control using unsupervised reinforcement learning, 2025. URL https://arxiv.org/abs/2511.04131.

[45] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graph.*, 37(4):143:1–143:14, July 2018. ISSN 0730-0301. doi:10.1145/3197517.3201311. URL http://doi.acm.org/10.1145/3197517.3201311.

[46] Y. Ze, S. Zhao, W. Wang, A. Kanazawa, R. Duan, P. Abbeel, G. Shi, J. Wu, and C. K. Liu. Twist2: Scalable, portable, and holistic humanoid data collection system, 2025. URL https://arxiv.org/abs/2511.02832.

[47] J. Li, X. Cheng, T. Huang, S. Yang, R.-Z. Qiu, and X. Wang. Amo: Adaptive motion optimization for hyper-dexterous humanoid whole-body control, 2025. URL https://arxiv.org/abs/2505.03738.

[48] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL https://arxiv.org/abs/2304.13705.