Fast Humanoid Loco-Manipulation via Flow Matching

Zehua Wang
MIT
zehuaw@mit.edu

Videet Mehta
MIT
mvideet@mit.edu

Qiao Sun
MIT
sqa24@mit.edu

***Abstract*—Humanoid loco-manipulation requires coordinated whole-body control for tasks like navigating while reaching for objects. Recent diffusion-based policies enable multi-task control through test-time guidance, but suffer from slow inference requiring 20–50 iterative denoising steps, limiting control frequencies to approximately 10 Hz. We address this bottleneck by using an alternative of DDPM in the field of computer vision: Flow Matching (FM), which enjoys both sample efficiency and simple implementation. Incorporating FM as the generative modeling achieves about 4× speedup while maintaining motion quality. FM achieves 820 survival steps using merely 5-NFE inference under perturbations, compared to DDPM's 280 survival steps, showing improvement in robustness at equivalent inference speed. Furthermore, we show that FM naturally supports guidance techniques as in DDPM counterpart. Notably, our training dataset contains *only walking motions* from LAFAN1, yet combined with training-free guidance, our model is able to conduct special tasks such as end-effector tracking (RMSE $\approx$ 1.1) and joystick velocity control (RMSE $\approx$ 0.55). This demonstrates outstanding generalizability from locomotion-only data to loco-manipulation behaviors without task-specific training. Our ablations reveal that longer prediction horizons are beneficial for guidance-based control, and adjusting guidance strengths can balance tracking accuracy with motion stability.**

***Index Terms*—humanoid robotics, loco-manipulation, flow matching, diffusion models, imitation learning**

I. Introduction

A. The Humanoid Loco-Manipulation Challenge

Humanoid robots represent a compelling platform for general-purpose automation in human-centered environments. Their anthropomorphic form factor enables them to navigate spaces designed for humans, manipulate objects at human-accessible heights, and perform tasks that leverage the full kinematic capabilities of the human body. However, realizing this potential requires solving the challenging problem of *loco-manipulation*: coordinating whole-body motion to achieve simultaneous locomotion and manipulation objectives.

Unlike fixed-base manipulators that can treat the arm as an isolated kinematic chain, humanoid manipulation fundamentally couples upper-body reaching with lower-body balance and locomotion. Reaching for an object shifts the robot's center of mass, requiring compensatory leg movements to maintain balance. Walking while carrying an object changes the system's inertial properties, affecting gait stability. These complex interactions demand policies that reason about whole-body coordination rather than treating locomotion and manipulation as separate problems.

The challenge is further compounded by the high dimensionality of humanoid state and action spaces. A typical humanoid has 20–30 actuated degrees of freedom, and generating coordinated motions across all joints while satisfying physical constraints (joint limits, torque limits, balance, collision avoidance) is computationally demanding. Traditional approaches using hand-crafted controllers or trajectory optimization are brittle and require extensive task-specific engineering. Recent learning-based methods train motion tracking policies via reinforcement learning on human motion capture data, achieving impressive tracking quality but requiring separate policies for each task [1].

A promising recent direction models trajectory generation as a conditional generative modeling problem, using diffusion models to learn distributions over feasible state-action trajectories. By predicting both future states and actions, these models can be steered toward task objectives through test-time guidance without retraining. However, synthesizing trajectories in real-time that simultaneously satisfy locomotion and manipulation objectives remains computationally prohibitive: current diffusion-based generative models require 20–50 denoising steps, each involving a full forward pass through a neural network. On edge-device hardware, this results in control frequencies around 10 Hz—too slow for highly dynamic maneuvers. Furthermore, hardware constraints limit diffusion policy effectiveness in humanoids, necessitating a faster, more efficient algorithm for denoising actions.

B. Limitations of Current Approaches

Recent work has demonstrated the versatility of diffusion-based trajectory generation for humanoid control. BeyondMimic [1] achieves zero-shot downstream task performance through test-time guidance, training Denoising Diffusion Probabilistic Models (DDPM) to predict joint state-action trajectories conditioned on observation history. By modeling both states and actions, the diffusion model can be steered toward task objectives without retraining. Despite its versatility, DDPM sampling requires 20–50 iterative denoising steps, each involving a full forward pass through the neural network. On edge-device hardware, this results in control frequencies around 10 Hz—too slow for highly dynamic loco-manipulation maneuvers like rapid reaching while walking or reactive corrections during arm movements.

Fig. 1: Example of robot waving it's hand from a Flow Matching policy using classifier guidance to wave its hand in the air

C. Our Approach and Contributions

This work addresses the inference speed bottleneck. We propose replacing DDPM with Flow Matching [2], [3], which has demonstrated superior performance in computer vision with both higher generation quality and faster sampling [4], [5]. Flow Matching learns continuous transport paths between noise and data distributions via ordinary differential equations, requiring significantly fewer function evaluations than discrete-time diffusion processes.

Building on the BeyondMimic paradigm, we implement a two-stage training pipeline: (1) train RL motion tracking policies on walking sequences from LAFAN1, (2) distill these experts into generative trajectory models via offline imitation, and (3) enable zero-shot downstream tasks through analytical test-time guidance. We make the following contributions:

1) **Flow Matching for faster inference**: We replace DDPM with Flow Matching to achieve 5–10× speedup while maintaining motion quality, enabling real-time control at higher frequencies.
2) **Zero-shot loco-manipulation from locomotion data**: We demonstrate that loco-manipulation capabilities can emerge from walking-only training data through test-time forward kinematics guidance, showing outstanding generalizability.

Our results demonstrate that Flow Matching enables fast and generalizable humanoid loco-manipulation control suitable for real-time applications.

II. Related Work

A. Traditional Robot Control

Classical robotic manipulation traditionally decouples global motion planning from local joint trajectories for fine-grained control. Sampling-based methods like Rapidly-exploring Random Trees (RRT) and their optimal variants (RRT*) [6] offer probabilistic completeness in high-dimensional configuration spaces, enabling path planning through complex obstacle fields. However, these methods plan in kinematic space and ignore dynamics, often producing trajectories that are infeasible for execution on physical systems with actuation limits and balance constraints.

Optimization-based planners like Graph of Convex Sets and libraries like TrajOpt directly optimize trajectories with respect to obstacles and collision costs [7]. These methods can incorporate dynamic constraints but are sensitive to initialization, computationally expensive for high-dimensional systems, and struggle with the non-convex constraints arising from contact dynamics.

For humanoid robots specifically, whole-body quadratic programming (QP) controllers have enabled impressive demonstrations of locomotion and manipulation [8]–[10]. These approaches formulate control as a prioritized optimization problem, balancing task objectives against physical constraints including joint limits, torque limits, and contact constraints. However, they require highly accurate dynamics models, carefully tuned cost functions, and extensive task-specific engineering. The resulting controllers are often brittle to model mismatch and struggle to generalize across diverse tasks.

For loco-manipulation specifically, traditional pipelines face fundamental limitations. The tight coupling between locomotion and manipulation means that planning must jointly optimize over the full body, dramatically increasing computational cost. Long-horizon coordination, where the robot must plan a sequence of locomotion and manipulation actions, remains particularly challenging. Moreover, deterministic planners produce single trajectories rather than distributions, limiting behavioral diversity and the ability to adapt to uncertainty.

B. Motion Tracking and Imitation Learning

Learning from human motion capture data offers an alternative to hand-designed controllers. The DeepMimic framework [11] demonstrated that reinforcement learning can train policies to track diverse motion capture clips, producing naturalistic, physically-feasible behaviors. Subsequent work has extended this paradigm to increasingly challenging motions and real hardware deployment.

Early motion-tracking approaches trained single-task policies for specific motions, requiring separate training runs for each behavior. Recent work has explored scalable frameworks that can track diverse motions within a single policy. PHC [12] demonstrated universal humanoid motion tracking in simulation, while OmniH2O [13], Exbody [14], and HumanPlus [15] achieved sim-to-real transfer for multi-motion tracking on physical hardware, though often with degraded motion quality compared to simulation results.

A key challenge in motion tracking is the gap between kinematic motion capture data and the actions required for physics-based execution. Motion capture provides joint angles and velocities but not the torques or position targets needed to reproduce those motions under physics simulation. Our approach addresses this by training per-trajectory tracking policies that learn to generate appropriate actions for each motion sequence.

C. Diffusion Models for Robot Control

Diffusion models have emerged as powerful tools for robot control, naturally handling the multi-modal action distributions that arise in complex tasks. Diffusion Policy [16] demonstrated that diffusion models can learn visuomotor policies for manipulation, achieving state-of-the-art performance on contact-rich tasks. DiffuseLoco [17] extended diffusion policies to quadruped locomotion, enabling smooth gait transitions and robust real-world deployment.

For humanoid control, BeyondMimic [1] represents the current state-of-the-art, training DDPM models on diverse LAFAN1 motion data and enabling zero-shot task performance through analytical test-time guidance. By jointly modeling states and actions, the diffusion model can be steered toward task objectives defined in state space without retraining. This approach achieves impressive versatility, performing waypoint navigation, joystick control, and obstacle avoidance with a single trained model. However, DDPM sampling requires 20–50 iterative denoising steps, limiting deployment to scenarios that can tolerate the resulting latency.

Diffuse-CLoC [18] introduced innovations for physics-based character control in simulation, demonstrating that joint state-action diffusion enables effective test-time guidance. PDP [19] showed that collecting data with noise injection during expert rollouts improves robustness by exposing the model to corrective actions from perturbed states.

D. Flow Matching for Generative Modeling

Flow Matching [2], [3] offers an alternative to diffusion models by learning continuous transport paths between distributions. Unlike DDPM's discrete-time reverse process, Flow Matching uses continuous ordinary differential equations, resulting in more efficient sampling. Recent work in computer vision [4], [5] has shown that Flow Matching achieves better generation quality with significantly fewer function evaluations.

The key advantage for robot control lies in the sampling efficiency: while DDPM requires 20–50 denoising steps, Flow Matching can generate high-quality samples in 5–10 steps. Recent work [4] has shown that classifier-free guidance transfers naturally from diffusion to Flow Matching, suggesting that analytical test-time guidance should also be compatible.

E. Imitation Learning and Distribution Shift

Imitation learning from offline demonstrations suffers from covariate shift [20]: policies trained on expert state distributions encounter novel states during deployment, leading to compounding errors. This is particularly problematic for loco-manipulation where deviations in whole-body coordination can cause task failure. While we focus on the inference speed problem in this work, addressing distribution shift through techniques like DAgger remains an important direction for future work.

F. Test-Time Guidance

Test-time guidance is a common strategy to steer generative models to certain tasks for task-consistent behavior without any additional supervision. In diffusion-based robot controllers, analytic constraints—such as target end-effector goals, contact schedules, or balance objectives—can be injected during sampling through classifier guidance (CG) or classifier-free guidance (CFG), effectively biasing the denoising process toward trajectories satisfying downstream control costs. Similar behavior is proven to work in Flow-Matching models where we transport samples through constraint-specified manifolds. For loco-manipulation specifically, this test-time guidance provides a way to incorporate goal-based tasks, feasibility constraints, and any physics priors that weren't learned by the generative model.

III. Method

Our approach follows a two-stage pipeline: (1) train motion tracking policies via reinforcement learning to generate action labels for motion capture data, and (2) distill these expert policies into a generative trajectory model that enables test-time guidance for downstream tasks.

A. Problem Formulation

Following BeyondMimic [1], we address the loco-manipulation control problem: given locomotion objectives (e.g., navigate to waypoint) and reaching objectives (e.g., move end-effector to position (x, y, z)), generate a physically feasible whole-body trajectory that simultaneously achieves both goals.

At each timestep t, our model predicts a trajectory $\boldsymbol{\tau}_t = [\mathbf{a}_t, \mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \ldots, \mathbf{s}_{t+H}, \mathbf{a}_{t+H}]$ containing the current action and H steps of future state-action pairs, conditioned on observation history $\boldsymbol{O}_t = [\mathbf{s}_{t-N}, \mathbf{a}_{t-N}, \ldots, \mathbf{s}_t]$ of length N.

B. Motion Tracking via Reinforcement Learning

Motion capture datasets provide kinematic references but lack action labels for imitation learning. Following BeyondMimic [1], we train motion tracking policies via reinforcement learning to generate actions that reproduce reference motions in physics simulation.

We train a separate policy π_i for each walking trajectory $\mathcal{T}_i$ in LAFAN1. This per-trajectory approach allows each policy to specialize without mode-averaging, enables parallel training, and achieves higher tracking fidelity than multi-task alternatives. Policies are trained using PPO with rewards based on pose and velocity tracking errors for selected body keypoints, plus regularization penalties for joint limits, action smoothness, and self-collision.

C. Flow Matching for Trajectory Generation

a) Training Objective: We train a prediction network $v_\theta(\boldsymbol{\tau}_t, t, \boldsymbol{O}_t)$ that predicts velocity targets for transporting trajectories from noise to data. By sampling a clean trajectory $\boldsymbol{\tau}_1$ and Gaussian noise $\boldsymbol{\tau}_0$, the training objective regresses on the conditional velocity:

$$\mathcal{L} = \mathbb{E}_{t,\boldsymbol{\tau}_0,\boldsymbol{\tau}_1}\left[\|v_\theta(\boldsymbol{\tau}_t, t, \boldsymbol{O}_t) - (\boldsymbol{\tau}_1 - \boldsymbol{\tau}_0)\|^2\right] \tag{1}$$

where v_θ is our network, $t \sim p(t)$ is a time schedule, and $\boldsymbol{\tau}_t$ is the noised trajectory at time t.

b) Test-Time Guidance: Following BeyondMimic, we enable zero-shot task performance through analytical guidance. For a task-specific cost function $G_{\text{task}}(\boldsymbol{\tau})$ (e.g., waypoint navigation, end-effector reaching), we guide the velocity output:

$$v'_\theta = v_\theta - w\nabla_{\boldsymbol{\tau}_t} G_{\text{task}} \tag{2}$$

where w controls guidance strength. This enables the same policy to perform diverse downstream tasks by simply changing the guidance objective at test time.

Following [21], we choose to not apply guidance on the final 5 denoising steps when using an NFE of 20. We empirically find that only applying guidance on an interval of steps greatly helps improve stability.

D. Forward Kinematics Guidance for Loco-Manipulation

We extend test-time guidance to loco-manipulation tasks without requiring manipulation-specific training data. Although our training data contains only walking motions, by defining guidance objectives over end-effector poses computed via forward kinematics, we can steer trajectories toward desired configurations:

$$\mathcal{L}_{\text{reach}} = \|\text{FK}(\mathbf{q}_t) - \mathbf{p}_{\text{target}}\|^2 \tag{3}$$

where FK maps joint angles to hand position. This enables zero-shot loco-manipulation: the same policy can perform reaching tasks by changing the guidance objective at test time.

E. Dataset Augmentation

A key insight from PDP [19] is that RL policies provide not only optimal trajectories but, more importantly, *corrective actions* from sub-optimal states. We leverage this insight by collecting **noisy-state clean-action** paired trajectories rather than standard clean-state clean-action pairs.

Specifically, during data collection we roll out each trained RL policy $\pi_{\mathcal{T}_i}$ with action noise injection: $\mathbf{a}_t = \pi_{\mathcal{T}_i}(\mathbf{s}_t, \tau) + \epsilon$, where ϵ is Gaussian noise. Critically, we store tuples of the form:

$$(\tilde{\mathbf{s}}_t,\ \mathbf{a}_t^\star,\ \mathbf{s}_{t+1}^\star),$$

where $\tilde{\mathbf{s}}_t$ is the *noisy state* resulting from previous noisy actions, but $\mathbf{a}_t^\star = \pi_{\mathcal{T}_i}(\tilde{\mathbf{s}}_t, \tau)$ is the *clean optimal action* from the expert policy. This creates a "noise band" around expert trajectories, enabling the model to learn corrective behaviors that recover from perturbations.

a) Multi-Scale Noise Augmentation: We extend the PDP framework by collecting rollouts at multiple noise scales simultaneously. For each motion sequence, we run 40 parallel environments with varying noise levels:

- 16 environments at $\sigma = 0.12$
- 12 environments at $\sigma = 0.16$
- 6 environments at $\sigma = 0.24$
- 6 environments at $\sigma = 0.36$

where σ controls the standard deviation of the Gaussian noise added to actions. This multi-scale approach widens the distribution coverage: lower noise levels ($\sigma = 0.12$) capture fine-grained corrections near the expert trajectory, while higher noise levels ($\sigma = 0.36$) expose the model to larger deviations requiring more aggressive recovery maneuvers. The graduated allocation (more environments at lower noise) ensures dense coverage near the expert while still sampling challenging recovery scenarios.

b) Final Dataset: The final dataset $\mathcal{D}$ is the union of all noise-augmented trajectories across all noise scales:

$$\mathcal{D} = \bigcup_{\sigma \in \{0.12, 0.16, 0.24, 0.36\}} \mathcal{D}_\sigma.$$

This augmentation strategy yields a generative model robust to state estimation errors and environmental perturbations while generating physically valid corrective actions.

IV. Experimental Setup

A. Data Collection and RL Training

We use the walking sequences from the LAFAN1 dataset. We train motion tracking policies using Proximal Policy Optimization (PPO) with adaptive sampling to emphasize challenging motion segments. For each motion sequence, we collect approximately 40 rollouts (≈3.5 hours of data) with action noise injection to perturb states and gather corrective actions, improving robustness.

a) State Representation via Forward Kinematics: We use the SMPL skeleton with $J = 23$ spherical joints. Rather than using joint-space representations directly, we transform generalized coordinates $(\mathbf{q}, \dot{\mathbf{q}})$ into Cartesian body positions and velocities via *forward kinematics* (FK). Given joint positions $\mathbf{q} \in \mathbb{R}^{3J}$ and velocities $\dot{\mathbf{q}} \in \mathbb{R}^{3J}$, FK computes the pose $T_b = \text{FK}_{\text{pos}}(\mathbf{q})$ and twist $\mathcal{V}_b = J_b(\mathbf{q})\dot{\mathbf{q}}$ for each body b, where J_b is the body Jacobian. This Cartesian representation offers two advantages: (1) it provides explicit spatial grounding for guidance objectives defined in task space, and (2) it avoids compounding errors that accumulate through the kinematic chain when predicting joint angles.

The state vector (165-dimensional) consists of:

- **Global States**: root position ($\mathbb{R}^3$), linear velocity ($\mathbb{R}^3$), and rotation ($\mathbb{R}^3$) as rotation vectors, expressed relative to the *character frame*—a local coordinate system with origin at the root, x-axis aligned with the character's facing direction, and z-axis pointing upward
- **Local States**: Cartesian body positions ($\mathbb{R}^{3J}$) and linear velocities ($\mathbb{R}^{3J}$) computed via FK, plus hand/ankle rotations ($\mathbb{R}^{3\times 4}$), all expressed in the character frame

b) Character Frame and Coordinate Transform: We explicitly define a character frame C. Let W denote the world frame and ${}^{W}p_{\text{root}} \in \mathbb{R}^3$ and q_{root} be the root (pelvis) position and orientation in W. We extract the yaw angle ψ from q_{root} and construct the yaw-only rotation

$$ {}^{W}R_C(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad {}^{C}R_W = ({}^{W}R_C)^\top. $$

The origin of C is at the root, ${}^{W}p_C = {}^{W}p_{\text{root}}$, with the x-axis aligned with the facing direction and z-axis pointing upward.

Given body positions ${}^{W}p_i$ and velocities ${}^{W}v_i$ in the world frame, and root linear velocity ${}^{W}v_{\text{root}}$, we obtain character-frame quantities via

$$ {}^{C}p_i = {}^{C}R_W\big({}^{W}p_i - {}^{W}p_{\text{root}}\big), \tag{4} $$

$$ {}^{C}v_i = {}^{C}R_W\big({}^{W}v_i - {}^{W}v_{\text{root}}\big). \tag{5} $$

All global and local state features described above are expressed in this character frame, providing translation and yaw invariance.

c) Action Space and PD Control: The action space (69-dimensional) represents *target joint positions* $\mathbf{q}^{\text{des}} \in \mathbb{R}^{3J}$ for a low-level proportional-derivative (PD) controller. The PD controller converts these position targets into joint torques:

$$ \boldsymbol{\tau} = K_p(\mathbf{q}^{\text{des}} - \mathbf{q}) - K_d\dot{\mathbf{q}}, \tag{6} $$

where $K_p, K_d \in \mathbb{R}^{3J\times 3J}$ are diagonal gain matrices for stiffness and damping, respectively. Following [1], we set gains based on reflected motor inertia: $k_{p,j} = I_j\omega_n^2$ and $k_{d,j} = 2I_j\zeta\omega_n$, where $\omega_n = 10$ Hz is the natural frequency, $\zeta = 2$ is the damping ratio (overdamped for stability), and $I_j = k_{g,j}^2 I_{\text{motor},j}$ is the reflected inertia accounting for gear ratio $k_{g,j}$.

The policy outputs normalized actions $\mathbf{a}_t \in [-1, 1]^{3J}$, which are scaled to target positions: $\mathbf{q}_t^{\text{des}} = \bar{\mathbf{q}} + \alpha\mathbf{a}_t$, where $\bar{\mathbf{q}}$ is a nominal configuration and α controls the action range. This PD formulation provides implicit torque information through current and prior commands while maintaining compliance for impact absorption—essential properties for dynamic humanoid control.

B. Model Training

We train two model variants to compare DDPM and Flow Matching:

1) **DDPM Baseline**: Standard DDPM following BeyondMimic
2) **Flow Matching**: Flow Matching with standard architecture

All models use a small Transformer decoder with 6 layers, 8 heads, and 512-dimensional embeddings (about 20M parameters, ≈1 GB GPU memory for inference). We train with observation history $N = 4$ (≈0.13s). By default, we use a predict horizon of 4 for FM and DDPM comparison and 16 for guidance-based control. We use the Adam optimizer with learning rate 1×10^{-3}, no weight decay, 10,000-step warmup, and constant learning rate schedule. We train models until convergence (which takes ≈24 hours) on 32 TPU v4 cores. Inference time is measured on an Nvidia RTX 4060 and A6000 GPU with TensorRT acceleration.

C. Evaluation Tasks and Metrics

We evaluate our models across multiple tasks to assess both speed and quality:

a) Locomotion Quality: We verify that Flow Matching achieves comparable tracking quality to DDPM on LAFAN1 sequences. We measure motion quality using Fréchet Inception Distance (FID) between generated and reference motions.

b) Loco-Manipulation via FK Guidance: We test zero-shot loco-manipulation through forward kinematics guidance:

- **End-effector waypoint precision**: Average Euclidean distance $\|\mathbf{p}_{\text{hand}}(t) - \mathbf{p}_{\text{target}}\|$ over trajectory, measured across 20 reaching trials with random target positions
- **Guidance composition success**: Percentage of trials where both locomotion goal (pelvis within 0.5m of waypoint) and reaching goal (hand within 15cm of target) are simultaneously achieved

V. Results

We evaluate our approach through three complementary experiments: (1) survival under perturbations comparing DDPM and Flow Matching at varying numbers of function evaluations (NFE), (2) joystick-based velocity control, and (3) end-effector position tracking for manipulation tasks.

A. Survival Comparison: Flow Matching vs. DDPM

We evaluate the intrinsic robustness of the two generative controllers by measuring their survival time under periodic perturbations. In this setting, the humanoid performs unconditional walking while receiving random root-velocity perturbations sampled uniformly from $[0, 0.5]$ m/s every second. Survival time is defined as the mean number of simulation steps before failure, where failure occurs when the head height drops below 0.2 m.

Figure 2 reports survival time across different NFEs on a logarithmic scale. At high NFE (20 steps), both models exhibit comparable robustness, achieving roughly 470 mean survival steps. As NFE decreases, however, a pronounced performance gap emerges. At NFE=5, Flow Matching retains strong stability with about 820 survival steps, while DDPM degrades to around 280 steps. This divergence further amplifies at lower NFEs: at NFE=2, Flow Matching achieves approximately 450 steps compared to DDPM's 150 steps; at NFE=1, Flow Matching still maintains around 280 steps, whereas DDPM collapses to roughly 30 steps. These results highlight Flow Matching's substantially superior robustness in low-NFE regimes.

This result has significant implications for real-time control. DDPM requires approximately 20 NFE to achieve stable walking, resulting in inference times of approximately 20ms

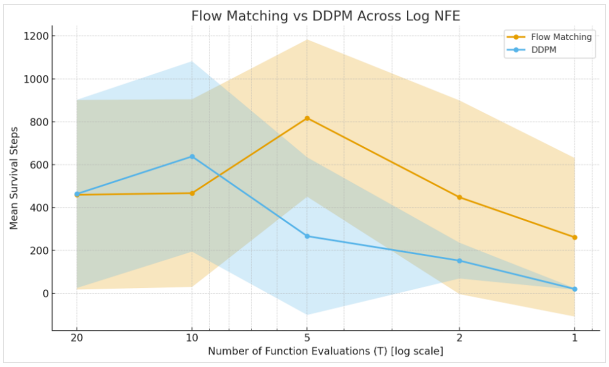

Fig. 2: Mean survival steps vs. number of function evaluations (NFE) for Flow Matching and DDPM. Flow Matching maintains robust performance even at very low NFE (1–5 steps), while DDPM degrades significantly below 10 steps. Shaded regions indicate standard deviation across 50 trials.

per control step. In contrast, Flow Matching achieves comparable or superior performance at NFE=5, enabling 4× faster inference. Qualitatively, we observe that DDPM at low NFE produces motions that deviate significantly from the training distribution, with unnatural poses and jerky movements, while Flow Matching generates smooth, natural walking even at NFE=1.

B. Joystick Velocity Control

We evaluate guided locomotion using joystick-style velocity commands. The guidance cost for joystick control is defined as the squared difference between predicted root velocity and the commanded velocity:

$$G_{\text{js}}(\boldsymbol{\tau}) = \frac{1}{2} \sum_{t'=t}^{t+H} \|\mathbf{v}_{xy,t'}(\boldsymbol{\tau}_{t'}) - \mathbf{g}_v\|^2 \tag{7}$$

where $\mathbf{v}_{xy,t'}$ extracts the planar root velocity at timestep t' and $\mathbf{g}_v \in \mathbb{R}^2$ is the goal velocity from the joystick controller. We issue a sequence of directional commands (forward, backward, turn left, turn right) each lasting 3 seconds and measure the root mean squared error (RMSE) between achieved and commanded velocities.

a) Effect of Prediction Horizon: Figure 3 (left) shows joystick RMSE as a function of prediction horizon H. We observe a consistent improvement with longer horizons: RMSE decreases from approximately 1.05 at $H = 2$ to 0.70 at $H = 16$. This 33% reduction demonstrates that longer prediction horizons enable more effective guidance, as the model can plan velocity changes over a longer temporal extent.

b) Effect of Guidance Scale w: Figure 4 (left) shows the effect of classifier guidance scale on joystick tracking performance. RMSE decreases from approximately 0.725 at $w = 10$ to 0.55 at $w = 40$, representing a 24% improvement. We observe a trade-off between control and stability when adjusting guidance scales: guidance scales above 40 begin to destabilize the walking gait, leading to increased fall rates. This suggests an optimal operating range of $w \in [30, 40]$ for velocity control tasks.

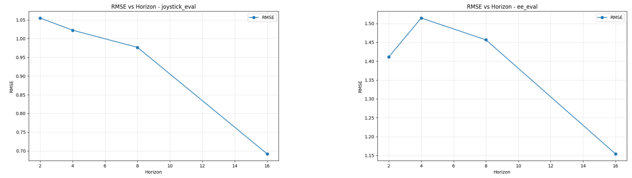

Fig. 3: RMSE vs. prediction horizon for joystick control (left) and end-effector tracking (right). Longer prediction horizons consistently improve tracking accuracy for both tasks, with RMSE decreasing by 33% and 24% respectively from $H = 2$ to $H = 16$.

C. Zero-Shot End-Effector Manipulation

The most striking result of our work is the emergence of manipulation capabilities from locomotion-only training data. Despite training exclusively on walking sequences from LAFAN1, our model combined with forward kinematics guidance achieves strong end-effector tracking performance.

The guidance cost for end-effector control is:

$$G_{\text{ee}}(\boldsymbol{\tau}) = \sum_{t'=t}^{t+H} \|\text{FK}(\mathbf{q}_{t'}) - \mathbf{p}_{\text{target}}\|^2 \tag{8}$$

where FK computes hand position from joint angles via forward kinematics. We evaluate reaching tasks where the humanoid must move its hand to randomly sampled target positions while maintaining stable walking.

a) Effect of Prediction Horizon: Figure 3 (right) shows end-effector RMSE versus prediction horizon. Similar to joystick control, longer horizons improve tracking: RMSE decreases from approximately 1.52 at $H = 2$ to 1.15 at $H = 16$, a 24% improvement. This confirms that the guidance mechanism benefits from extended temporal planning for both locomotion and manipulation objectives.

b) Effect of Classifier Guidance Scale w: Figure 4 (right) reveals an interesting non-monotonic relationship between guidance scale and end-effector tracking. RMSE decreases from 1.49 at $w = 20$ to a minimum of approximately 1.11 at $w = 100$, then slightly increases at higher guidance scales. This indicates an optimal guidance strength that balances tracking accuracy with motion naturalness—too weak guidance fails to reach targets, while too strong guidance produces unnatural arm movements that compromise overall stability.

D. Generalization from Locomotion to Manipulation

An interesting finding of this work is that manipulation capabilities emerge naturally from locomotion-only training data when combined with test-time guidance. Our training dataset contains exclusively walking motions—no reaching, grasping, or manipulation demonstrations. Yet the model successfully tracks end-effector targets with reasonable accuracy (RMSE ≈ 1.1 at optimal settings).

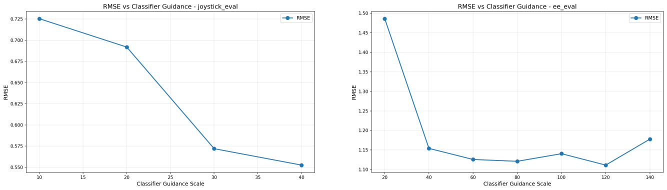

Fig. 4: RMSE vs. classifier guidance scale for joystick control (left) and end-effector tracking (right). Both tasks show improved tracking with stronger guidance up to an optimal point, after which stability begins to degrade.

This generalization occurs because: (1) walking motions contain natural arm swings and postural adjustments that span a substantial portion of the arm's reachable workspace, (2) the Flow Matching model learns a smooth manifold of physically plausible poses that can be continuously deformed toward novel configurations, and (3) classifier guidance provides the steering signal to navigate this manifold toward task-relevant regions.

This finding has significant practical implications: it suggests that manipulation capabilities can be bootstrapped from more readily available locomotion datasets, reducing the need for expensive manipulation-specific motion capture or teleoperation data collection.

VI. Discussion

A. *Why Flow Matching Outperforms DDPM at Low NFE*

Our results demonstrate that Flow Matching significantly outperforms DDPM when operating at low numbers of function evaluations—precisely the regime required for real-time robot control. We attribute this to fundamental differences in how the two approaches model the generative process.

DDPM learns a discrete-time reverse diffusion process where each denoising step attempts to remove a fixed amount of Gaussian noise. When the number of steps is reduced, the model must make larger "jumps" per step, and errors accumulate rapidly because each step's prediction depends on the previous step's output. The noise schedule in DDPM is carefully calibrated for a specific number of steps, and deviating from this schedule leads to distribution mismatch.

In contrast, Flow Matching learns a continuous velocity field that directly transports samples from noise to data along straight paths. The optimal transport formulation encourages straighter trajectories, which can be accurately integrated with fewer discretization steps. Even with a single Euler step (NFE=1), Flow Matching produces samples that lie approximately on the data manifold, whereas DDPM at NFE=1 produces samples that are far from the learned distribution.

This theoretical advantage translates directly to practical benefits for humanoid control: Flow Matching enables 4–10× faster inference while maintaining or improving motion quality, enabling control frequencies that approach the requirements for dynamic loco-manipulation.

B. *The Horizon-Guidance Trade-off*

Our ablation studies reveal complementary roles for prediction horizon and guidance strength. Longer horizons improve task performance (lower RMSE) by enabling the model to plan trajectories that smoothly satisfy guidance objectives over extended time periods. This is particularly important for manipulation tasks where the hand must traverse significant distances while maintaining balance.

However, longer horizons also increase computational cost linearly with H. Combined with our finding that Flow Matching achieves good performance at low NFE, this suggests an optimal operating point: use moderate horizons ($H \approx 16$) with few integration steps (NFE ≈ 5) to achieve both good task performance and real-time inference.

Guidance strength exhibits a different trade-off: stronger guidance improves task accuracy but can destabilize the underlying motion. We observe that excessively strong guidance pushes the model toward poses that are out-of-distribution relative to the training data, resulting in physically implausible configurations and eventual falls. The optimal guidance strength varies by task—end-effector manipulation requires stronger guidance (CG $\approx$ 100) than velocity control (CG $\approx$ 40)—likely because manipulation targets are farther from the natural arm positions in walking data.

C. *Limitations*

Our approach has several limitations that suggest directions for future work:

a) Limited Motion Diversity: Our training data contains only walking motions, which constrains the range of achievable behaviors. While manipulation capabilities emerge through guidance, the model cannot produce motions that differ substantially from walking, such as crouching, jumping, or turning in place. Expanding the training data to include more diverse locomotion styles would likely improve both the range and quality of guided behaviors.

b) Guidance Tuning: The optimal guidance strength varies across tasks and must be tuned empirically. An automatic guidance adaptation mechanism that adjusts strength based on task difficulty or current tracking error would improve usability.

c) Distribution Shift: Like all imitation learning approaches, our method suffers from distribution shift when the policy encounters states not seen during training. The noise-augmented data collection partially addresses this, but compounding errors can still lead to degraded performance over long rollouts. Techniques like DAgger [20] that collect additional data from the deployed policy could improve robustness.

d) Skill Transitions: The model struggles to transition between distinct behaviors. When guidance objectives change suddenly (e.g., switching from walking forward to reaching a distant target), the model may become trapped in its current mode rather than smoothly transitioning to a new behavior. This is a fundamental limitation of the diffusion/flow matching approach when the target behavior lies far from the current state on the learned manifold.

VII. Future Work

Several promising directions extend this work:

a) Real-World Deployment: Our experiments are conducted in simulation. Deploying on physical hardware requires addressing additional challenges including state estimation noise, communication latency, and unmodeled dynamics. The robustness demonstrated in our perturbation experiments suggests the policy may transfer well, but systematic sim-to-real experiments are needed.

b) Multi-Task Training: Training on more diverse motion data—including running, jumping, dancing, and manipulation demonstrations—would expand the range of achievable behaviors. The per-trajectory training approach scales naturally to larger datasets.

c) Hierarchical Control: Combining our approach with higher-level planning could enable more complex tasks. A planner could generate waypoint sequences that the Flow Matching policy tracks, enabling navigation in cluttered environments or multi-step manipulation tasks.

d) Online Adaptation: Incorporating online learning mechanisms like DAgger would address distribution shift by collecting corrective data during deployment. This is particularly important for long-horizon tasks where compounding errors become significant.

VIII. Conclusion

We presented a fast humanoid loco-manipulation system that replaces DDPM with Flow Matching for efficient trajectory generation. Our key findings are:

1) **Efficient inference via Flow Matching**: Flow Matching achieves 4× faster inference than DDPM while maintaining or improving motion quality. At NFE=5, Flow Matching achieves 820 mean survival steps compared to DDPM's 280 steps—demonstrating that the efficiency gains do not sacrifice robustness. This enables control frequencies suitable for dynamic humanoid maneuvers.
2) **Outstanding generalizability from locomotion to manipulation**: Despite training exclusively on walking motions from LAFAN1, our model combined with classifier guidance achieves strong end-effector tracking (RMSE $\approx$ 1.1) and velocity control (RMSE $\approx$ 0.55). This demonstrates that loco-manipulation capabilities can emerge from locomotion-only data through test-time forward kinematics guidance, eliminating the need for expensive manipulation-specific data collection.
3) **Prediction horizon improves guidance**: Longer prediction horizons consistently improve task performance, with RMSE reductions of 24–33% when increasing horizon from 2 to 16 steps. This suggests that effective guidance requires planning over extended temporal windows.

Our results establish Flow Matching as a superior alternative to DDPM for real-time humanoid control, particularly in low-latency regimes where DDPM's performance degrades significantly. The emergence of manipulation capabilities from locomotion data suggests a path toward building versatile humanoid controllers from readily available motion datasets. Future work will focus on real-world deployment, expanding training data diversity, and incorporating online adaptation to address distribution shift over long-horizon tasks.

Acknowledgment

We thank the 6.4210 teaching staff for their guidance and feedback throughout this project.

References

[1] Q. Liao, T. E. Truong, X. Huang, G. Tevet, K. Sreenath, and C. K. Liu, "Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion," 2025. [Online]. Available: https://arxiv.org/abs/2508.08241

[2] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," 2023. [Online]. Available: https://arxiv.org/abs/2210.02747

[3] X. Liu, C. Gong, and qiang liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow," in *The Eleventh International Conference on Learning Representations*, 2023.

[4] N. Ma, M. Goldstein, M. S. Albergo, N. M. Boffi, E. Vanden-Eijnden, and S. Xie, "Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers," *arXiv preprint*, vol. arXiv:2401.08740, 2024. [Online]. Available: https://arxiv.org/abs/2401.08740

[5] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel *et al.*, "Scaling rectified flow transformers for high-resolution image synthesis," in *Forty-first international conference on machine learning*, 2024.

[6] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, p. 846–894, Jun. 2011. [Online]. Available: https://doi.org/10.1177/0278364911406761

[7] T. Marcucci, J. Umenberger, P. A. Parrilo, and R. Tedrake, "Shortest paths in graphs of convex sets," *arXiv preprint arXiv:2101.11565*, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2101.11565

[8] L. Saab, O. E. Ramos, F. Keith, O. Stasse, P. Souères, and A. Cherubini, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *IEEE Transactions on Robotics*, vol. 29, no. 2, p. 346–362, 2013. [Online]. Available: https://doi.org/10.1109/TRO.2012.2234351

[9] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, vol. 2, no. 4, p. 505–518, 2005. [Online]. Available: https://doi.org/10.1142/S0219843605000586

[10] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas robot in the DARPA robotics challenge," *Autonomous Robots*, vol. 40, no. 3, p. 429–455, 2016. [Online]. Available: https://doi.org/10.1007/s10514-015-9479-3

[11] 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.*, vol. 37, no. 4, pp. 143:1–143:14, Jul. 2018. [Online]. Available: http://doi.acm.org/10.1145/3197517.3201311

[12] Z. Luo, J. Cao, A. W. Winkler, K. Kitani, and W. Xu, "Perpetual humanoid control for real-time simulated avatars," in *International Conference on Computer Vision (ICCV)*, 2023.

[13] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi, "Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning," 2024. [Online]. Available: https://arxiv.org/abs/2406.08858

[14] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang, "Expressive whole-body control for humanoid robots," 2024. [Online]. Available: https://arxiv.org/abs/2402.16796

[15] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn, "Humanplus: Humanoid shadowing and imitation from humans," 2024. [Online]. Available: https://arxiv.org/abs/2406.10454

[16] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” 2024. [Online]. Available: https://arxiv.org/abs/2303.04137

[17] X. Huang, Y. Chi, R. Wang, Z. Li, X. B. Peng, S. Shao, B. Nikolic, and K. Sreenath, “Diffuseloco: Real-time legged locomotion control with diffusion from offline datasets,” 2024. [Online]. Available: https://arxiv.org/abs/2404.19264

[18] X. Huang, T. Truong, Y. Zhang, F. Yu, J. P. Sleiman, J. Hodgins, K. Sreenath, and F. Farshidian, “Diffuse-cloc: Guided diffusion for physics-based character look-ahead control,” *ACM Transactions on Graphics*, vol. 44, no. 4, p. 1–12, Jul. 2025. [Online]. Available: http://dx.doi.org/10.1145/3731206

[19] T. E. Truong, M. Piseno, Z. Xie, and K. Liu, “Pdp: Physics-based character animation via diffusion policy,” in *SIGGRAPH Asia 2024 Conference Papers*, ser. SA ’24. ACM, Dec. 2024, p. 1–10. [Online]. Available: http://dx.doi.org/10.1145/3680528.3687683

[20] S. Ross, G. J. Gordon, and J. A. Bagnell, “A reduction of imitation learning and structured prediction to no-regret online learning,” 2011. [Online]. Available: https://arxiv.org/abs/1011.0686

[21] T. Kynkäänniemi, M. Aittala, T. Karras, S. Laine, T. Aila, and J. Lehtinen, “Applying guidance in a limited interval improves sample and distribution quality in diffusion models,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 122 458–122 483, 2024.