Generating Physically Realistic and Directable Human Motions from Multi-Modal Inputs

Aayam Shrestha[*1], Pan Liu[*2], German Ros[†2], Kai Yuan[‡2], and Alan Fern[1]

[1] Oregon State University
[2] Intel Labs

Abstract. This work focuses on generating realistic, physically-based human behaviors from multi-modal inputs, which may only partially specify the desired motion. For example, the input may come from a VR controller providing arm motion and body velocity, partial key-point animation, computer vision applied to videos, or even higher-level motion goals. This requires a versatile low-level humanoid controller that can handle such sparse, under-specified guidance, seamlessly switch between skills, and recover from failures. Current approaches for learning humanoid controllers from demonstration data capture some of these characteristics, but none achieve them all. To this end, we introduce the Masked Humanoid Controller (MHC), a novel approach that applies multi-objective imitation learning on augmented and selectively masked motion demonstrations. The training methodology results in an MHC that exhibits the key capabilities of *catch-up* to out-of-sync input commands, *combining* elements from multiple motion sequences, and *completing* unspecified parts of motions from sparse multimodal input. We demonstrate these key capabilities for an MHC learned over a dataset of 87 diverse skills and showcase different multi-modal use cases, including integration with planning frameworks to highlight MHC's ability to solve new user-defined tasks without any finetuning. Project webpage: `https://panwliu.github.io/mhc/`

Keywords: Humanoid Motion Generation · Multimodal · Imitation

1 Introduction

Physically simulated humanoid characters have the potential to generate natural looking and realistic human motions and behaviors for a variety of applications ranging from video games, robotics, virtual/augmented reality, to digital avatars. However, directable and natural motion generation for these physically unstable, high-dimensional characters corresponds to a challenging control problem requiring precise coordination of joint-level commands [14,28]. One solution is to leverage motion capture (MoCap) data, which provides detailed pose information for

---

∗ Equal contribution

† Now at NVIDIA

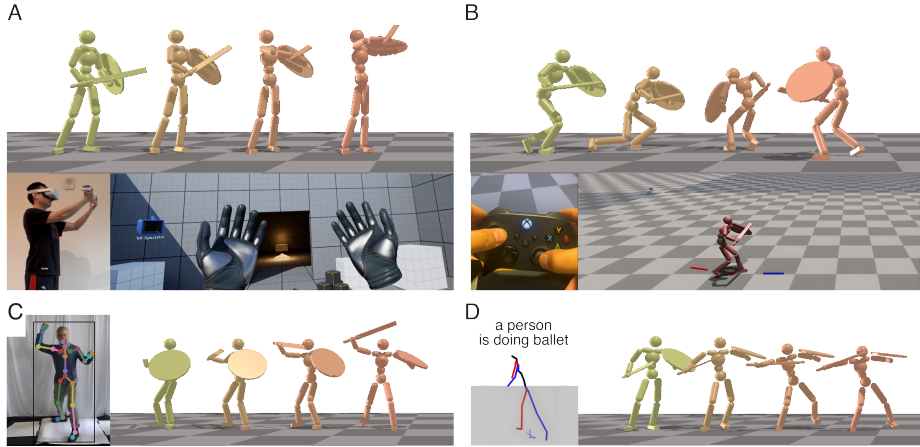‡ Corresponding author. Email: kai.yuan@intel.com

Fig. 1: Showcases generated human motions from multi-modal inputs: (A) VR device, (B) joystick controller, (C) video, and (D) text. Our proposed method, Masked Humanoid Controller (MHC), can generate physically realistic motions from a wide variety of muli-modal directives.

demonstrations of real human motion that are relevant to an application. We can then attempt to train a motion generator that can be directed to imitate the individual motion demonstrations as well as combinations or transformations of these motions. This data-driven imitation-learning approach offers the potential for producing directable behavior while ensuring human-like motion quality.

A key aim of our work is to train a motion generator that can be directed through multiple input modalities such as video, text, and VR controllers as well as high-level goal specifications, without the need for fine-tuning. As an example, given limited demonstrations of *running* and *stationary waving*, we should be able to generate motions for a new behavior "running while waving" by directing the generator with high-level waypoints for navigation along with arm motion directives. Importantly, such waypoint directives are sparse in that they do not specify the joint-level details of the desired motion. Thus, the generator must be able to automatically fill-in-the-blanks to achieve natural looking motions that satisfy the sparse directives. In addition, the generator must be able to recover from failures, smoothly switch between different motions, and blend motions into novel combinations. We identify these crucial capabilities as: ***Catch-up***, ***Combine*** and ***Complete*** - (CCC) (Fig. 2).

The *Catch-up* capability is required to support transitioning between different motion directives. This refers to resynchronizing to the directed motion from states that are currently inconsistent with that motion, e.g., failure states. This allows initiating motions from arbitrary fallen poses and seamlessly transitioning between different reference motions mid-way through execution. The *Combine* capability involves imitating novel combinations of upper and lower body movements by blending segments from different motion demonstrations; for instance, walking while waving. Finally, the *Complete* capability allows for sparse, under-specified guidance signals; such as missing certain joint information, by tracking
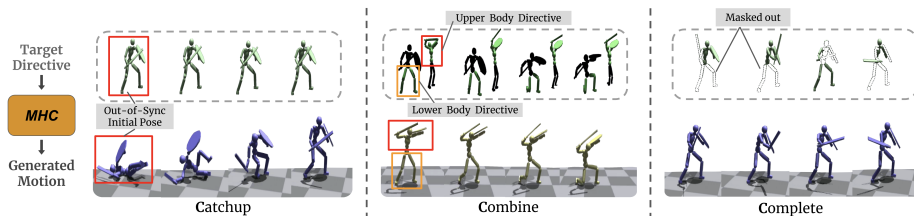
Fig. 2: Shows generated motions that illustrate the CCC capabilities. From left to right: MHC is able to generate motions that (1) adjust and *catchup* starting from an out-of-sync pose, (2) imitate a target directive that *combines* upper and lower body sub-segments from different motions, and (3) *complete* the motion from under-specified directives as indicated by missing target outlines.

available cues and implicitly filling in missing details. Together, these capabilities enable multimodal directives across MoCap, text, video, joystick and VR. Additionally, they facilitate defining finite state machines to generate motions using high level user-specifications, as well as integration with data-driven planners to generate motions that achieve user-specified high-level goals.

To enable the CCC capabilities under a unified framework, we present the Masked Humanoid Controller (MHC). In summary, our main contributions are three-fold: (1) We identify three key missing capabilities in current humanoid controllers - catch-up, combine and complete - that are jointly-necessary for adaptable behaviors, (2) We propose the Masked Humanoid Controller (MHC) - a novel multi-objective imitation learning framework that utilizes augmented and masked demonstrations to enable these capabilities, (3) We integrate MHC with planning frameworks, enabling complex goal-driven behaviors using finite state machines (FSMs) and data-driven planners (DAC-MDPs [25]). The CCC capabilities are extensively validated by training an MHC on a diverse dataset of 87 motor skills [22]. Additionally, we showcase MHC's zero-shot planning capabilities using a wide range of finite state machines as well as case studies on integration with DAC-MDPs for motion generation adhering to high-level goals.

2   Related Work

Learning from MoCap Data: Leveraging MoCap data enables controllers to acquire complex behaviors with human-like motion quality. However, large and diverse MoCap datasets present challenges including scalable distillation and ensuring fluid transition between skills. Initial works train single-clip policies to mimic individual behaviors using tracking rewards [4,19,23] or adversarial losses [7, 11, 17, 21, 32]. However, distilling these into a multi-clip controller remains computationally prohibitive [28]. A more scalable alternative is to directly learn a multi-clip policy learning via reinforcement learning with tracking objectives [5, 14, 16, 20, 28, 30, 31]. However, tracking rewards alone is not enough to ensure smooth transitions between skills and failure recovery. Recent works augment training with adversarial losses to encourage natural motions during transitions [20, 27] or define an explicit fail state recovery policy [14].
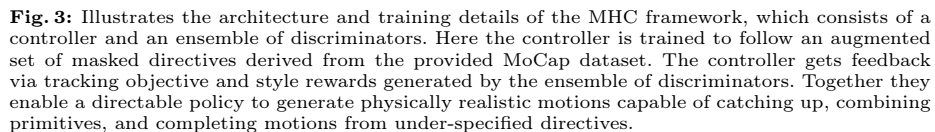
Combination of motions: Recent works have explored imitating combined motions, but require training individual policy for each new behavior pair [1, 12, 33]. Additionally, they rely on full motion oversight, lacking adaptability to partial guidance. On the completion front, some kinematic models can synthesize motions despite missing information [3, 29]. However, these controllers are not grounded in physics, restricting their application.

Under-specified Control: Intuitive modalities like language, video, and VR provide important yet often under-specified means to direct motor skills. Existing works map some of these modalities to embedding spaces [10, 11, 26, 34] or key joint poses [13, 14]. However, they handle a predefined sparisty types; adapting to new sparsity specifications like VR require expensive retraining [2,6]. While ambiguity is inherent in language and image-conditioned policies [8, 34], fine-grained control remains difficult as they do not allow low-level granularities like joint-level guidance. Overall a gap persists in controllers that can handle partial, sparse guidance with precision across modalities. Closing this gap can enable more intuitive control of reusable motor behaviors.

Downstream Planning: The acquired low-level control skills can support downstream tasks via a wide range of approaches including supervised fine-tuning for specialized behaviors [34], reinforcement learning for new objectives [5, 13, 18, 20], model predictive control for short-term horizons [9], and finite state machines that encode behavioral logic [27]. However, supervised fine-tuning remains restricted in flexibility to new tasks while reinforcement learning lacks sample efficiency whereas model predictive formulations are limited to short planning horizons. Additionally, the range of possible finite state machines largely depend on the flexibility of the underlying low-level controller. One solution is the use of data-driven planners like DAC-MDPs [25] which compile static experiences into approximated MDPs for fast optimization. While these methods enable zero-shot generalization, their integration with learned reusable motor skills remains relatively unexplored. Overall, leveraging low-level controllers to swiftly accomplish high-level goals remains an open challenge.

3   Problem Formulation

Given a dataset of human motion demonstrations, our objective is to develop a motion generator that can produce directable, realistic combinations of those and similar motions in simulated physics environments. By directable we mean that the generator can be provided with directives that specify properties of the desired future motion. For example, the most detailed directive is to specify the exact future positions and velocities of each humanoid joint. Alternatively an under-specified directive might provide just the desired hand positions in the future time window or just the torso velocity. For such under-specified directives, the motion generator should produce motion that is consistent with the provided directive, while also constraining to the space of natural motions. By allowing for under-specified directives we can support many input modalities for describing

Fig. 3: Illustrates the architecture and training details of the MHC framework, which consists of a controller and an ensemble of discriminators. Here the controller is trained to follow an augmented set of masked directives derived from the provided MoCap dataset. The controller gets feedback via tracking objective and style rewards generated by the ensemble of discriminators. Together they enable a directable policy to generate physically realistic motions capable of catching up, combining primitives, and completing motions from under-specified directives.

motion. For example, MoCap data provides fully-specified directives whereas joystick input may only specify the root velocity and arm motion.

More formally, a *motion* is a sequence $q_{1:H}$ of multi-channel poses for a humanoid with J joints. Each pose is a tuple $q_i = (q^r, q^\theta, q^l, q^g)$, where q^r specifies the position, orientation, linear and angular velocity of the root joint, $q^\theta \in \mathbb{R}^{J \times 6}$ [3] is the 3D joint rotations, $q^l \in \mathbb{R}^{J \times 3}$ is the local joint positions relative to the root pose, and $q^g \in \mathbb{R}^{J \times 3}$ is the global joint position in the world coordinates.

Motion directives are used to specify constraints on a desired motion to be generated. Specifically, in this work, a *motion directive* d is defined as a masked motion sequence represented by $d = (\hat{q}_{1:H}, I_{1:H})$, where $\hat{q}_{1:H}$ is a motion and $I_{1:H}$ is a sequence of binary masks such that I_i indicates which dimensions of pose q_i are selected as motion constraints. For example, a directive that only specifies future hand positions would select only the dimensions corresponding to hand positions in $\hat{q}^l$.

In this work, we focus on generating motions that are dynamically consistent with a physics-simulation environment. In particular, it is not sufficient to simply generate kinematic pose sequences that "look right". For this purpose we consider motion generators that dynamically control the actuators of physically simulated humanoids. Specifically, as shown in Fig. 3 our motion generator takes as input the pose of the humanoid q_t at time t and the current motion directive $(\hat{q}_{t+1:t+H}, I_{t+1:t+H})$, which specifies the desired motion for the next H steps. The output of the generator a_t gives the set points for each joint of the simulated humanoid. These set points are then provided to a PD controller with fixed gains, which produces actuator torques for the physics simulator. Since,

---

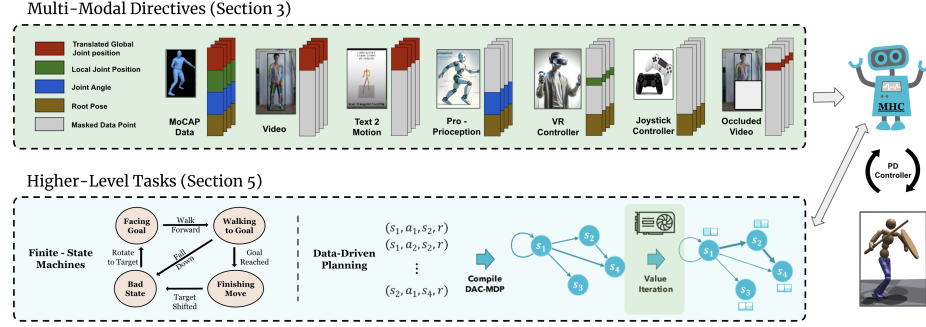[3] We use the 6 DoF rotation representation for orientation [20].

Fig. 4: Highlights the potential applications of MHC. [Top] The selective masking of the target directive allows MHC to represent various modalities of motion data under a single framework. These multi-modal inputs include MoCap, full or occluded video, joystick, VR controller among others. [Bottom] Similarly selective masking of target directive also allows us to treat the guiding signal itself as abstract actions. This enables straightforward integration with Finite State Machines and Data Driven Planning to allow zero-shot motion generation for higher-level task specifications.

our motion generator controls a physical system and is conditioned on masked motion directives, we call it the Masked Humanoid Controller (MHC).

In order to learn an MHC we assume access to a dataset of reference motions $\mathcal{M}$, which encompasses the types of motions that are needed for the application goals. Our goal is to learn an MHC capable of producing motions that are combinations and in-plane rotations of motions in $\mathcal{M}$. Specifically, we consider combinations of upper and lower body motions in $\mathcal{M}$ and denote the augmented dataset of all such combinations as $\mathcal{M}^+$. Note, however, that our learning framework can be instantiated for any meaningful type of combination. In addition, the MHC should be able to smoothly switch between motion sub-segments in $\mathcal{M}^+$, even when switching to an out-of-sync motion, which requires "catching up". Note that ideally, the learned generator will generalize beyond motions in $\mathcal{M}^+$, but the degree of generalization is primarily an empirical question.

Finally, the learned MHC should be able to produce motions based on different multi-modal sources of motion data. Each type of source will correspond to a particular type of directive whose mask identifies the motion constraints provided by that source. Figure. 4 illustrates the types of inputs and corresponding directive masks we consider in this work. For example, given human motion video, tracking software can be used to generate a directive $(\hat{q}_{1:H}, I_{1:H})$ such that $\hat{q}$ only has components for $\hat{q}^g$ and I selects only those components. Additionally, when joints are occluded, I only selects the un-occluded joints.

4   Masked Humanoid Controller

Because the MHC involves solving a complex physical control problem without supervision for the low-level actions, we formulate MHC training as reinforcement learning (RL). Training proceeds through episodes, where each episode initializes the humanoid in a physical environment with a specific directive. The

reward during an episode indicates how well the motion matches the target directive and reflects natural motion.

4.1 Training Episodes

Our goal is to create a distribution over episodes that will require the MHC to switch between motions in $\mathcal{M}^+$ based on directives corresponding to multimodal inputs. If an MHC achieves high reward on such a distribution, then it will have the desired CCC properties. Each episode is defined by a randomly sampled pair of initial humanoid pose and target directive.

Initial Pose Distribution: The initial pose distribution is a mixture of a uniform distribution over poses in $\mathcal{M}^+$ and a distribution of fallen humanoid poses $p_{fall}(q)$, where the weight of p_{fall} is 0.1 in our experiments. This choice forces the MHC to learn catchup to out-of-sync target directives, including recovery from fallen humanoid states. In addition, we apply a random in-plane rotation when generating a pose so that the MHC is robust to orientation changes.

Target Directive Distribution: For each episode we generate a target directive $(\hat{q}_{1:L}, I_{1:L})$, where L is the length of the episode. The motion $\hat{q}_{1:L}$ for the directive is generated by concatenating random length subsequences drawn from $\mathcal{M}^+$. In our experiments, $L = 300$, corresponding to 10 seconds, and each sub-sequence length is uniform in the range from 120 to 240. This type of concatenation generally results in sharp and inconsistent motion transitions, forcing the MHC to learn catchup behavior. In addition, we also apply a random in-plane rotation to each sub-sequence so that the MHC is robust to orientation changes.

The directive mask $I_{1:L}$ for an episode is generated by generating a mask I_1 for the first time step and using that mask for all other time steps (i.e. $I_t = I_1$ for all t). We employ two types of masking: channel-level and joint-level masking (Fig. 4). For, channel level masking, one or more of the available channels $(\hat{q}^r, \hat{q}^\theta, \hat{q}^l, \hat{q}^g)$ are selectively masked. The different channel combinations that we randomly choose between each episode are detailed in Fig. 4. For example, all channels except $\hat{q}^g$ can be masked to represent the joint keypoint modality that can be generated from video via computer vision. Channel level masking allows us to represent modalities like video tracking, text2motion models, proprioception, and joystick controllers.

Joint-level masking is a secondary level of masking that can be further applied to the joint position channels (q^l, q^g). Here we sample a percentage in $[0, 100]$ and randomly select that percentage of joints to mask out from the target directive. This allows us to emulate partially occluded video motions as well as VR controllers which only have a limited number of keypoint sensors. In our experiments we sample masks by first randomly selecting a channel level mask followed by a 50% chance of also applying joint-level masking.

4.2 Reward Design

The RL training objective is to maximize the expected discounted episodic reward denoted by $\mathbb{E}\left[\sum_{t=1}^{L} \gamma^{t-1} r_t\right]$ where $\gamma \in (0, 1]$ is the discount factor and r_t

is the reward at timestep t. We define the reward at each step as the sum of a tracking reward r_t^{tr}, which encourages agreement with directives, a style reward r_t^{st}, which encourages natural looking motions, and an additional energy cost c_t, which encourages smooth motions: $r_t = 0.5 r_t^{tr} + 0.5 r_t^{\text{st}} - c_t$.

Tracking Reward: The tracking reward is defined to prefer generated motions based on how well they agree with the episode's directive at each time step. We find that learning is accelerated by using a reward function that prioritizes learning coarse motion characteristics before focusing on finer motion details. In particular, we define four reward terms in order of priority $r_t^{\text{h}}, r_t^{\text{o}}, r_t^{\text{v}}, r_t^{\text{l}}$ corresponding to matching the directive for root height, root orientation, root velocity and joint Euler coordinates, respectively. At each time step, a term is activated only if the higher priority rewards are above a threshold of 0.9. Specifically, the tracking reward function is given by: $r_t^{tr} = r_t^{\text{h}} + r_t^{\text{o}} + r_t^{\text{v}} + r_t^{\text{l}}$

$$r_t^h = e^{-m_h \cdot 8 ||q_t^h - \hat{q}_t^h||_2} \tag{1}$$

$$r_t^o = I(r_t^h > 0.9) \cdot e^{-m_o \cdot ||d(q_t^o - \hat{q}_t^o)||_2} \tag{2}$$

$$r_t^v = I(r_t^o > 0.9) \cdot e^{-m_v \cdot ||q_t^v - \hat{q}_t^v||_2} \tag{3}$$

$$r_t^l = I(r_t^v > 0.9) \cdot \frac{1}{\sum_{j \in J} m_j} \sum_{j \in J} e^{-m_j \cdot 40 ||q_t^j - \hat{q}_t^j||_2} \tag{4}$$

where J is the set of all joints, and m_h, m_o, m_v, m_j are equal to 0 if the corresponding reward component depends on the pose information not selected by the episode's directive mask, and otherwise equal to 1. Thus, reward terms not relevant to the directive do not affect preferences over the generated motion.

Style Reward: We consider a multi-part style reward, inspired by [1], with a distinct discriminator for different body parts. Specifically, we create 5 sets of joints from the whole body: $\mathcal{J}_1 \mathcal{J}_2, \mathcal{J}_3, \mathcal{J}_4,$ and $\mathcal{J}_5$ corresponding to upper right, upper left, root, lower body, and full body, respectively. During RL training a discriminator D_{ϕ_k} is trained for each part set $\mathcal{J}_k$, conditioning only on joint information in $\mathcal{J}_k$. Training is done continuously throughout RL by sampling a pair of motions, one positive motion from $\mathcal{M}$ and one negative motion generated by the current MHC, and updating the parameters of each discriminator using the regularized loss function from [1,21]. In our implementation, the input to the discriminators is a motion sub-sequence of length 10 so that the discriminator takes temporal dependencies into account. The corresponding reward component at time step t for discriminator k is $-\log(1 - D_{\phi_k}(q_{t-10:t}^k))$. The overall style reward r^{st}, is the average of these components across discriminators.

Energy Cost: To penalize large changes in the action across timesteps and large torques, the energy cost at time t is given by

$$c_t = \sum_{j \in J} 0.01 \cdot ||a_t^j - a_{t-1}^j||_1 + 0.0002 \cdot ||\tau_t^j||_1 \tag{5}$$

where a_t^j is the action (set point) at time t for joint j and τ_t^j is the torque applied to joint j. The inclusion of this penalty is critical for avoiding high-frequency jitter, especially of the foot.
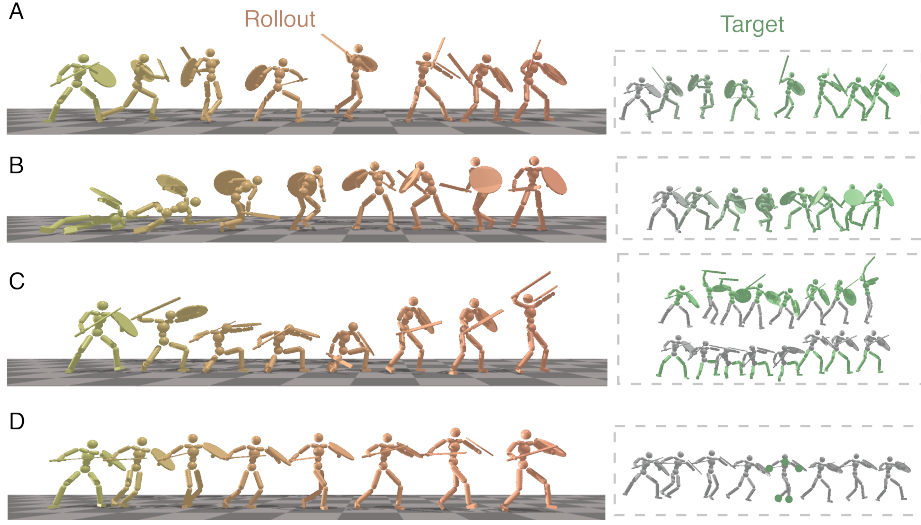
Fig. 5: Illustrates generated motions corresponding to key CCC capabilities of MHC. The simulation (left) displays key-frames of humanoid following different motion directives (right). From top to bottom the simulated humanoid (A) follows an imitation target, (B) transitions from falling-down position to catch-up to the target directive, (C) imitates motion directive that combines upper-body and lower-body movements from distinct motions (D) completes the motion using only 3D joint positions of the head, hands and feets.

5   Motion Generation for Higher-Level Tasks

We propose to integrate the learned MHC with a data-driven planning framework that allow for automatically generating a potentially large FSM for a user-defined higher-level task. Instead of specifying the FSM directly, the designer specifies: 1) a reward function r^* and optimization objective (e.g. discounted total reward) corresponding to the higher-level task, 2) the set of possible directives A for the MHC that can be used as dynamically selected actions to achieve the task, and 3) a mask over pose variables S defining a state abstraction used to form the FSM states. For example, the user may specify a reward based on the distance to a goal location, specify directives corresponding to root velocity and orientation commands, and a state abstraction corresponding to localized goal location. Alternatively if certain types of arm motions were required, directives and the state abstraction could include selected sub-motions for the arm joints.

We follow the data-driven planning framework of DAC-MDPs [25] which uses data collected from a dynamic system to construct and solve a Markov Decision Process (MDP) as shown in 4. In our application, we collect trajectory data by randomly initializing the MHC in the environment and executing random sequences of directives from A each trajectory stores the state abstraction at each time step along with the selected action directive. The DAC-MDP solution is a control policy that specifies for each abstract state, which directive in A to execute. We refer the reader to [25] for DAC-MDP details. Importantly, the CCC properties of the MHC allow it to robustly execute the high-level actions selected

Table 1: Results for the ASE baseline, the MHC, and MHC ablations for the imitation, catchup and combine experiments. The ablation differs across expeirments. For imitation we remove the style reward, for catchup we remove random pose initialization, and for combine we remove target motion augmentation (i.e. training on $\mathcal{M}$ rather than $\mathcal{M}^+$. The MHC outperforms the ASE baseline for all tasks across the Reallusion dataset $\mathcal{M}_{Real}$ and ASE rollout dataset $\mathcal{M}_{ASE}$. The ablation experiment demonstrate that each component is crucial and jointly enable the CCC capabilities.

Method	Imitation				Catchup				Combine			
	$\mathcal{M}_{Real}$		$\mathcal{M}_{ASE}$		$\mathcal{M}_{Real}$		$\mathcal{M}_{ASE}$		$\mathcal{M}_{Real}$		$\mathcal{M}_{ASE}$	
	$mpjpe \downarrow$	$Suc \uparrow$	$mpjpe \downarrow$	$Suc \uparrow$	$mpjpe \downarrow$	$Suc \uparrow$	$mpjpe \downarrow$	$Suc \uparrow$	$mpjpe \downarrow$	$Suc \uparrow$	$mpjpe \downarrow$	$Suc \uparrow$
ASE [20]	123.51	0.6	100.28	0.77	125.96	0.55	102.5	0.7	210.75	0.17	197.94	0.25
MHC (Ours)	**51.05**	**0.92**	**56.23**	**0.97**	**59.24**	**0.89**	**63.46**	**0.98**	**95.09**	**0.48**	**60.95**	**0.78**
MHC (abl)	552.25	0.0	555.7	0.0	66.23	0.3	87.1	0.17	103.73	0.44	69.68	**0.78**

by the DAC-MDP. It is worth noting that this approach is zero-shot in the sense that no fine-tuning or RL training is required to address a new high-level task. Rather, this framework uses efficient optimal planning to produce solutions.

6   Experiments

We conduct experiments to empirically validate MHC's key capabilities to imitate a wide range of motion directives that necessitate the key capabilities of imitation, catchup, combination and completion (Fig. 5). We then showcase motion generation from target directives derived from multiple modalities (Fig. 7) and case studies on motion generation for higher-level tasks. (Fig. 8).

Datasets and Baselines: We train our MHC on a Reallusion MoCap dataset [20,22] of 87 motion clips ($\mathcal{M}_{Real}$). We also generate a test set $\mathcal{M}_{ASE}$ of 87 motions using a pre-trained ASE controller [20] by initializing it in a random pose and selecting a sequence of random ASE skills. We also use this pre-trained ASE controller as a state-of-the-art baseline for the Reallusion dataset. Specifically, ASE includes an encoder that maps target motions to skills, which allows us to evaluate ASE on fully specified directives. Note that, unlike MHC, the ASE controller does not support under-specified directives. Thus, in our comparisons, we always provide ASE with directives that specify all pose information.

Metrics In our experiments we evaluate how well a generated motion matches the specified motion directive. As a metric of motion similarity we use the commonly used mean per-joint position error E_{MPJPE} (in mm), which averages the root relative error of each humanoid joint. Inspired by UHC [15], we also measure the success rate ($Succ$) defined as following the reference motion with $< 10\%$ of failed frames, where a failed frame is one where maximum joint error is > 1m. For under-specified directives that mask joints, the success rate and E_{MPJPE} is measured only using the selected joints.

6.1   CCC Capabilities

Imitation: We measure the imitation quality of the MHC and ASE baselines across both datasets $\mathcal{M}_{Real}$ and $\mathcal{M}_{ASE}$. Figure 5A illustrates an example of
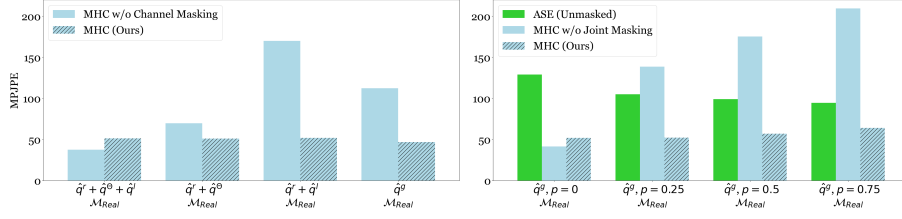
Fig. 6: (Left) Performance across different channel-level masks. We find that the MHC trained with directive masking retains its imitation performance across different variants of channel masks in contrast to the MHC trained without masking. (Right) Performance across different percentages of joint masking (0% to 75%). We see that the MHC shows stable performance as the amount of masking increases compared to the MHC trained without joint-level masking. We also see that the MHC significantly outperforms ASE, even though ASE is provided fully-specified (unmasked) directives. The performance of ASE varies across masking levels because we only evaluate the metric over unmasked joints.

the MHC imitating a target directive. To assess imitation performance, we use fully-specified target directives corresponding to the training and testing motions. The initial humanoid pose is set to match the initial pose of the target directive and then the controllers generate the motion using a sliding window lookahead of the target directive. As presented in Table 1, MHC demonstrates superior performance compared to ASE on both datasets, as evidenced by the lower E_{MPJPE} and higher success rate metrics. These results suggest that the MHC exhibits better imitation fidelity compared to the ASE baseline. We also consider an MHC ablation where we remove the sytle reward during training, using only tracking rewards and energy costs. Table 1 shows that the ablated MHC, denoted MHC(abl), fails to learn effective motion generation, even from these fully-specified directives. This finding underscores the critical role played by the style reward term in the successful training of MHC.

Catchup: To assess the catchup capabilities we initialize the MHC or ASE in a random pose drawn from the training or testing sets. We then run the motion generator with a directive that is derived by concatenating a pair of random subsequences from the training or testing sets. Figure 5B showcases an example of the MHC successfully catching up to a target motion from a randomly initialized fallen pose, highlighting its adaptability and resilience. Table 1 presents a comparative analysis of MHC against ASE baseline and an ablation trained without random pose initialization. The evaluation is performed across both the training and test datasets, with the MHC consistently achieving superior E_{MPJPE} and success rates compared to the baselines. Note that since catchup inherently involves non-synchronization with the directive, we loosen the success criteria so that a trial is considered successful if $< 25\%$ of the frames fail. These results provide strong evidence for MHC's enhanced capability in failure recovery and smooth catchup transitions, even when starting from perturbed states.

Combine: We now evaluate the ability to handle directives that combine upper and lower body joint subsets, as outlined in Section 4. Here the target directive is generated by randomly sampling a pair of motion trajectories from either the training or testing sets and combining the upper and lower body movements
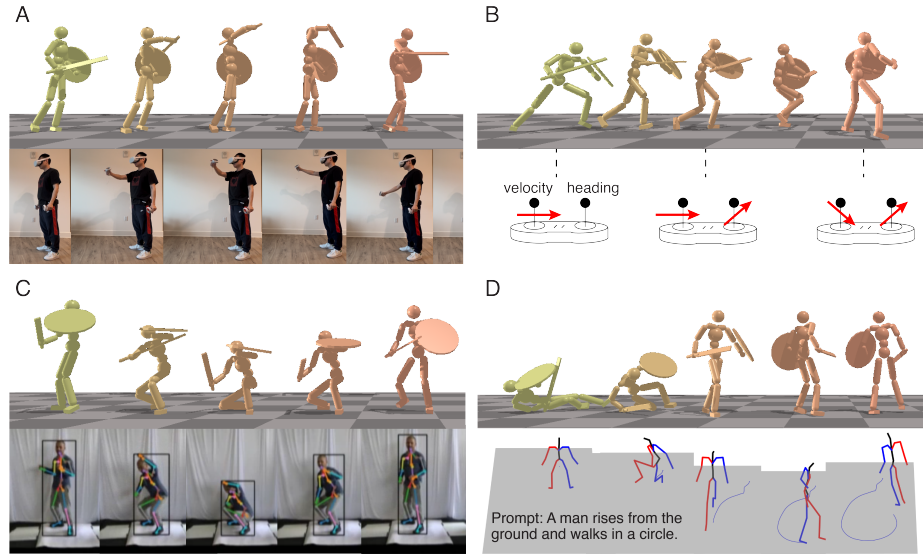
Fig. 7: Illustrates qualitative results using keyframes for motion generation under multi-modal inputs such as (A) VR headset and controllers, (B) joystick controllers, (C) 3D joint positions derived from video and (D) text-to-motion generator. This highlights the versatility of MHC and its applications for motion generation directed by various modalities that may be noisy, under-specified.

from each motion. Figure 5C illustrates an example of the MHC successfully imitating a target directive derived by combining upper and lower-body movements from distinct motion sequences. Table 1 presents the quantitative results for imitating combined motions, where the motion pairs are sampled from $\mathcal{M}_{Real}$ and $\mathcal{M}_{ASE}$. We see that the MHC consistently outperforms both the ASE baseline and an ablated version of MHC. Here the ablation is trained without data augmentation i.e. $\mathcal{M}$ instead of $\mathcal{M}^+$. The ablation highlights the importance of data augmentation in enabling MHC to effectively work with directives that combine different body sub-segments from distinct motions.

Complete: Finally, we evaluate MHC's ability to generate motions from under-specified target directives. We do this by repeating the setup for imitation task, this time with under-specified directives. As described in Section 3, different choices of under-specification relate to different motion-data modalities. We start by evaluating channel level masks were we specify one of the following: full pose information ($\hat{q}^r + \hat{q}^\theta + \hat{q}^l$), pro-prioception ($\hat{q}^r + \hat{q}^\theta$), pro-prioception with forward kinematics ($\hat{q}^r + \hat{q}^l$), and global joint keypoint positions $\hat{q}^g$. Figure 6 (left) shows that MHC can perform well for these channel level under-specification of directives. In a second experiment, we consider joint-level masking where we progressively mask 0% to 75% of the joints in $\hat{q}^g$. These directives correspond to potentially occluded keypoint trajectories from Video. Figure 6 (right) shows that the MHC can retain its performance and generate motions with significantly lower E_{MPJPE} as compared to the ASE baseline. This is notable since ASE is provided with all of the joint information, since it does not support
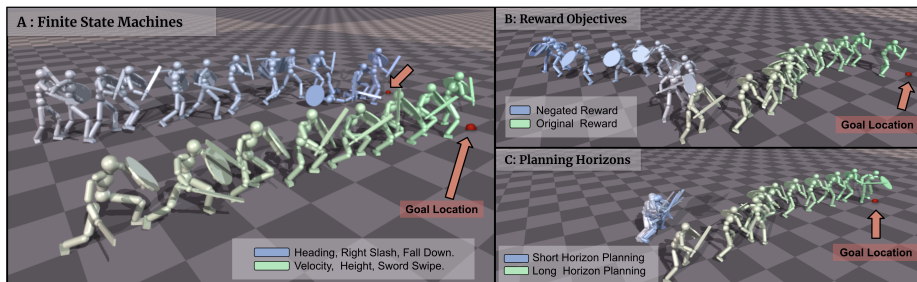
Fig. 8: (A) Key frame visualization of FSMs for Go-To-Location task. FSM A1 (blue) generates a motion of walking towards the goal position while doing right sword slashes and ultimately falling down and recovering on reaching the goal. FSM A2 (green) generates a motion that swings its sword as it crouch-walks towards the goal while facing the other away. and swinging the sword along the way. (B) Visualization of FSM produced by DAC-MDP using original/negated reward functions for Go-To-Location task. FSM B1 (green) uses original reward function and reliably reaches the goal. FSM B2 (blue) uses negated reward function and avoids the goal. (C) Visualization of FSM produced by DAC-MDP for different discount factors. FSM C1 (blue) uses small discount factor and plans for short horizon. The generated motion swings the sword to get the immediate reward followed by episode termination. FSM C2 (green) use larger discount factor and plans for long horizon. The generated motion moves towards the goal to collect larger long-term reward.

under-specified directives. Finally, we condider an MHC ablation where we train without joint-level masking. We see that the ablated version performs poorly as the amount of masking increase. Figure 5D illustrates an example of the MHC successfully imitating an under-specified target motion where only the 3D positions for the head, feet, and hands are provided. This highlights MHC's ability to generate realistic motion even from a diverse set of under-specified directives.

6.2   Multi-Modal Directives

We qualitatively evaluate the MHC for different kinds of real-world input modalities for motions that the MHC was never trained on. We explore the following modalities: virtual reality devices, joystick controllers, 3D joint positions extracted from videos, and text-to-motion generation. In our experiments, we utilized the Meta Quest 2 VR system, comprising a headset and two controllers, to gather the orientation and 3D positions of both the headset and controllers (Fig. 7A). For joystick-based inputs, an Xbox controller was employed to manipulate the speed and direction of the velocity (via the left joystick and trigger), root orientation and height (via the right joystick and trigger), as illustrated in Fig. 7B. The estimation of 3D poses from video footage was achieved using MeTRAbs [24] (Fig. 7C), while generation of kinematic human motions from textual descriptions was achieved using T2M-GPT [35] (Fig. 7D). We find that the learned MHC is versatile and can generate natural motions from under-specified directives derived from these input modalities. These results have important implications for real-world applications, where motion capture data may often be incomplete or noisy due to occlusions or sensor limitations.

6.3   Higher-level Task Specification

We provide demonstrations of motion generation for higher-level tasks via integration of the MHC with FSMs and DAC-MDPs. For FSM integration we consider the task of navigating to a specified target goal location while executing different locomotion styles, such as running or crouch-walking, and performing various finishing moves, like taunting or falling down as shown in Figure 8A. We hand-coded simple FSMs to achieve these tasks, where each state of the FSM selects an appropriate directive for the MHC to execute. This includes under-specified directives corresponding to joystick commands and a fully-specified directives corresponding to the finishing moves.

We integrate the MHC with DAC-MDPs by constructing a DAC-MDP using a dataset collected by generating motions using the MHC with directives derived from random joystick commands. The first high-level task (Figure 8B) is to simply reach a goal location. To do this, we define a DAC-MDP reward function that provides a reward at each step that is larger as the humanoid gets closer to the goal. The reward at the goal is a maximum of 1. The resulting FSM produced by solving this DAC-MDP results in motions that reliably reach the goal. To illustrate the zero-shot capabilities, we next give the DAC-MDP the negation of the first reward function, which should lead the agent to avoid the goal location. Figure 8B illustrates the resulting desired behavior.

Finally, we illustrate the higher-level reasoning capabilities afforded via the DAC-MDP integration. As illustrated in Figure 8C, we adjust the reward function and DAC-MDP termination condition so that the humanoid receives a positive reward for swinging the sword, which immediately terminates the episode. We then use the DAC-MDP to produce an FSM for a large discount factor 0.999 and a smaller discount factor 0.9. The large/small discount factors encourage planning over a long/short horizon. The best short horizon plan is to simply get the immediate reward of swinging the sword and then ending the episode, while the long-horizon plan is to instead go to the goal and collect the larger long-term reward. This is exactly the behavior produced by the two FSMs.

7   Summary

We highlighted three capabilities, Catchup, Combine, and Complete, that are jointly necessary for practical applications of physically-realistic motion generation. Our proposed MHC is the first motion generator that achieves all three capabilities. This RL-based approach for training is highly versatile and can be applied to any available motion-capture dataset and multi-modal input directives. Importantly, the MHC is a dynamic motion generator in the sense that it actively generates motion in response to the current environment conditions. Finally, we demonstrated a straightforward integration of the MHC with data-driven planning to allow for zero-shot motion generation for higher-level tasks.

Acknowledgements

References

1. Bae, J., Won, J., Lim, D., Min, C.H., Kim, Y.: Pmp: Learning to physically interact with environments using part-wise motion priors. ACM SIGGRAPH 2023 Conference Proceedings (2023)
2. Cerón, J.C., Sunny, M.S.H., Brahmi, B., Mendez, L.M., Fareh, R., Ahmed, H.U., Rahman, M.H.: A novel multi-modal teleoperation of a humanoid assistive robot with real-time motion mimic. Micromachines **14** (2023)
3. Chen, X., Jiang, B., Liu, W., Huang, Z., Fu, B., Chen, T., Yu, J., Yu, G.: Executing your commands via motion diffusion in latent space. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 18000–18010 (2022)
4. Chentanez, N., Müller, M., Macklin, M., Makoviychuk, V., Jeschke, S.: Physics-based motion capture imitation with deep reinforcement learning. Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games (2018)
5. Dou, Z., Chen, X., Fan, Q., Komura, T., Wang, W.: C·ase: Learning conditional adversarial skill embeddings for physics-based characters. ArXiv **abs/2309.11351** (2023)
6. Du, Y., Kips, R., Pumarola, A., Starke, S., Thabet, A.K., Sanakoyeu, A.: Avatars grow legs: Generating smooth human motion from sparse tracking inputs with diffusion model. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 481–490 (2023)
7. Ho, J., Ermon, S.: Generative adversarial imitation learning. In: Neural Information Processing Systems (2016)
8. Huang, Z., Zhou, Z., Chuang, Y.Y., Wu, J., Liu, C.K.: Physically plausible animation of human upper body from a single image. 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) pp. 930–939 (2022)
9. zhengyao jiang, Xu, Y., Wagener, N., Luo, Y., Janner, M., Grefenstette, E., Rocktäschel, T., Tian, Y.: H-GAP: Humanoid control with a generalist planner. In: NeurIPS 2023 Foundation Models for Decision Making Workshop (2023)
10. Juravsky, J., Guo, Y., Fidler, S., Peng, X.B.: Padl: Language-directed physics-based character control. SIGGRAPH Asia 2022 Conference Papers (2022)
11. Lee, C.H., Yueh, N.C.L., Woo, K.T.: Human-inspired video imitation learning on humanoid model. 2022 Sixth IEEE International Conference on Robotic Computing (IRC) pp. 345–352 (2022)
12. Lee, S., Lee, J., Lee, J.: Learning virtual chimeras by dynamic motion reassembly. ACM Transactions on Graphics (TOG) **41**, 1 – 13 (2022)
13. Luo, Z.G., Cao, J., Merel, J., Winkler, A., Huang, J., Kitani, K., Xu, W.: Universal humanoid motion representations for physics-based control. ArXiv **abs/2310.04582** (2023)
14. Luo, Z., Cao, J., Winkler, A., Kitani, K., Xu, W.: Perpetual humanoid control for real-time simulated avatars. `https://doi.org/10.48550/arXiv.2305.06456`
15. Luo, Z., Yuan, Y., Kitani, K.M.: From universal humanoid control to automatic physically valid character creation. ArXiv **abs/2206.09286** (2022)
16. Merel, J., Hasenclever, L., Galashov, A., Ahuja, A., Pham, V., Wayne, G., Teh, Y.W., Heess, N.M.O.: Neural probabilistic motor primitives for humanoid control. ArXiv **abs/1811.11711** (2018)

17. Merel, J., Tassa, Y., Dhruva, T., Srinivasan, S., Lemmon, J., Wang, Z., Wayne, G., Heess, N.M.O.: Learning human behaviors from motion capture by adversarial imitation. ArXiv **abs/1707.02201** (2017)
18. Merel, J., Tunyasuvunakool, S., Ahuja, A., Tassa, Y., Hasenclever, L., Pham, V., Erez, T., Wayne, G., Heess, N.: Catch & carry: Reusable neural controllers for vision-guided whole-body tasks
19. Peng, X.B., Abbeel, P., Levine, S., Van de Panne, M.: Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. ACM Transactions On Graphics (TOG) **37**(4), 1–14 (2018)
20. Peng, X.B., Guo, Y., Halper, L., Levine, S., Fidler, S.: Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. ACM Transactions On Graphics (TOG) **41**(4), 1–17 (2022)
21. Peng, X.B., Ma, Z., Abbeel, P., Levine, S., Kanazawa, A.: Amp: Adversarial motion priors for stylized physics-based character control. ACM Transactions on Graphics (ToG) **40**(4), 1–20 (2021)
22. Reallusion: 3d animation and 2d cartoons made simple.
23. Ren, J., Yu, C., Chen, S., Ma, X., Pan, L., Liu, Z.: DiffMimic: Efficient motion mimicking with differentiable physics
24. Sárándi, I., Linder, T., Arras, K.O., Leibe, B.: MeTRAbs: metric-scale truncation-robust heatmaps for absolute 3D human pose estimation. IEEE Transactions on Biometrics, Behavior, and Identity Science **3**(1), 16–30 (2021). `https://doi.org/10.1109/TBIOM.2020.3037257`
25. Shrestha, A., Lee, S., Tadepalli, P., Fern, A.: Deepaveragers: Offline reinforcement learning by solving derived non-parametric mdps. ArXiv **abs/2010.08891** (2020)
26. Sun, J., Zhang, Q., Duan, Y., Jiang, X., Cheng, C., Xu, R.: Prompt, plan, perform: Llm-based humanoid control via quantized imitation learning. ArXiv **abs/2309.11359** (2023)
27. Tessler, C., Kasten, Y., Guo, Y., Mannor, S., Chechik, G., Peng, X.B.: Calm: Conditional adversarial latent models for directable virtual characters. ACM SIGGRAPH 2023 Conference Proceedings (2023)
28. Wagener, N., Kolobov, A., Frujeri, F.V., Loynd, R., Cheng, C.A., Hausknecht, M.: MoCapAct: A multi-task dataset for simulated humanoid control. `https://doi.org/10.48550/arXiv.2208.07363`
29. Wang, J., Xu, H., Xu, J., Liu, S., Wang, X.: Synthesizing long-term 3d human motion and interaction in 3d scenes. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 9396–9406 (2020)
30. Won, J., Gopinath, D.E., Hodgins, J.K.: A scalable approach to control diverse behaviors for physically simulated characters. ACM Transactions on Graphics (TOG) **39**, 33:1 – 33:12 (2020)
31. Won, J., Gopinath, D.E., Hodgins, J.K.: Physics-based character controllers using conditional vaes. ACM Transactions on Graphics (TOG) **41**, 1 – 12 (2022)
32. Xu, P., Karamouzas, I.: A gan-like approach for physics-based imitation learning and interactive control. Proceedings of the ACM on Computer Graphics and Interactive Techniques **4**, 1 – 22 (2021)
33. Xu, P., Shang, X., Zordan, V.B., Karamouzas, I.: Composite motion learning with task control. ACM Transactions on Graphics (TOG) **42**, 1 – 16 (2023)
34. Yao, H., Song, Z., Zhou, Y., Ao, T., Chen, B., Liu, L.: Moconvq: Unified physics-based motion control via scalable discrete representations. ArXiv **abs/2310.10198** (2023)

35. Zhang, J., Zhang, Y., Cun, X., Zhang, Y., Zhao, H., Lu, H., Shen, X., Shan, Y.: Generating human motion from textual descriptions with discrete representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14730–14740 (2023)