





Physics-Based Motion Tracking of Contact-Rich Interacting Characters

Xiaotang Zhang¹ , Ziyi Chang¹ , Qianhui Men² , and Hubert P. H. Shum^{†1} 

¹Durham University, United Kingdom
{xiaotang.zhang, ziyi.chang, hubert.shum}@durham.ac.uk

²University of Bristol, United Kingdom
qianhui.men@bristol.ac.uk

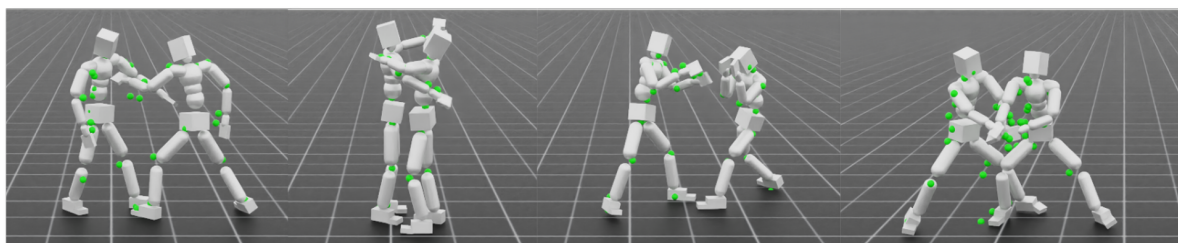


Figure 1: Physics-based motion tracking of two humanoid characters performing contact-rich interactions such as boxing, pushing, and grappling. The objective is to track and reproduce stable motions under frequent physical contacts and complex force exchanges.

Abstract

Motion tracking has been an important technique for imitating human-like movement from large-scale datasets in physics-based motion synthesis. However, existing approaches focus on tracking either single character or a particular type of interaction, limiting their ability to handle contact-rich interactions. Extending single-character tracking approaches suffers from the instability due to the challenge of forces transferred through contacts. Contact-rich interactions requires levels of control, which places much greater demands on model capacity. To this end, we propose a robust tracking method based on progressive neural network (PNN) where multiple experts are specialized in learning skills of various difficulties. Our method learns to assign training samples to experts automatically without requiring manually scheduling. Both qualitative and quantitative results show that our method delivers more stable motion tracking in densely interactive movements while enabling more efficient model training.

Keywords: animation system, physical simulation, motion tracking

CCS Concepts

• Computing methodologies → Physical simulation; Motion capture; Motion processing;

1. Introduction

Physics-based motion tracking enables the synthesis of physically valid movements by using the next-frame pose as a control signal. Despite extensive research, existing approaches focus exclusively on either single-character motion tracking [PALV18; LCKX*23], or a particular type of interactions with a task-specific controller

[WGH21; ZZLH23; LWL*24], limiting their ability to handle contact-rich interactions.

Extending single-character motion tracking methods to interaction settings faces a fundamental challenge: preserving stability while transferring forces through contacts. In physical simulation, identical poses may have different underlying contact forces that may vary significantly. When characters are controlled independently by single-character tracking methods without modeling force transfer, contact-rich interactions become unstable and may

[†] Corresponding author

even fail to track, as one character may receive inconsistent and uncontrolled forces from its opponent. For example, different poses in highfive have different distributions of contact forces between hands. Single-character tracking is unable to handle this since it does not take potential transferred forces into account. When sudden forces are transmitted through contacts, they can lead to oscillations or unmeaningful movements.

Since contact-rich interactions like boxing or dancing introduce physical perturbations, another challenge of this task is that they place much greater demands on model capacity. This added complexity significantly increases the risk of catastrophic forgetting in tracking network [LCKX*23]. Prior work on physics-based humanoid interactions [ZZLH23; WGH21; LWL*24] usually addresses this by restricting the problem scope: training skill-specific policies [WGH21; PALV18], adopting multi-stage learning paradigms [ZZLH23], or focusing on sparse and long-range interactions [LWL*24]. However, these strategies either incur high training costs or compromise performance on the dense, contact-rich interaction details.

To overcome these challenges, we propose a progressive, all-in-one mixture-of-experts architecture. Intuitively, interactions naturally involve different levels of control complexity. For example, in boxing, low-level control maintains stable locomotion such as stance and stepping, while mid-level to high-level control governs reactive behaviors such as dodging and blocking. These controls operate hierarchically, with higher-level strategies relying on the stability of low-level control. Building on this intuition, our method eliminates the need for multi-stage skill-specific training [LCKX*23; PALV18] by introducing a progressive mixture-of-experts design, enabling a single policy for contact-rich interactions.

Specifically, we introduce a progressive training strategy for the mixture-of-experts architecture, drawing inspiration from the progressive neural network (PNN) framework [RRD*16]. This strategy addresses the instability in robot states caused by dense contacts in contact-rich interactions. By routing samples based on tracking error, the system automatically assigns contact-free, high-reward, stable samples to base experts and contact-heavy, low-reward, unstable samples (where force transfer occurs and is challenging to track) to specialized experts. In contrast to the original PNN where expert policies are manually assigned to distinct datasets or tasks, our approach removes the need for such predefined partitioning. By training directly on the entire motion dataset, the framework autonomously allocates samples of varying difficulty to the most appropriate experts.

We demonstrate our method on InterHuman [LZL*24], achieving robust tracking accuracy and smooth transfer across different interaction patterns. We also validate the stability of our approach under perturbations introduced via obstacles or next-frame poses. Finally, we analyze the contributions of individual experts, highlighting the model's ability to capture different levels of torque control. Our contributions are summarized as follows:

- *All-in-one framework* We propose an architecture for physics-based motion tracking in contact-rich interactions, removing the reliance on task-specific or multi-stage controllers.

- *Progressive mixture-of-experts* We introduce a progressive mixture-of-experts strategy that gradually adding new experts to model the hierarchical levels of torque control, enabling stable prediction of joint torques.
- *Robust interaction tracking* We demonstrate the effectiveness of our method on large-scale datasets, showing superior tracking accuracy, smooth and realistic interactions across diverse tasks, and strong robustness under perturbations.

2. Related Works

2.1. Physics-based Humanoid Motion Tracking

Since no ground-truth data exist of human joint actuation and physics simulators are often non-differentiable, a policy, aka controller, is often trained to track and mimic human motion using deep reinforcement learning (RL). From [PALV18], RL-based motion tracking has gone from imitating single clips to large-scale datasets [CMM*18; WGSF20; LCKX*23; FBH21]. Among them, a mixture of experts [WGH20], differentiable simulation [RYC*23], and external forces [YK20] have been used to improve the quality of motion imitation. Recently, Luo et al. [LCKX*23] allows a single policy to mimic almost all of AMASS and recover from falls. Luo et al. [LCM*23] improve [LCKX*23] to track all AMASS and distill its motor skills into a latent space. Luo et al. [LCC*24] focuses on humanoid motion imitation with articulated fingers. Luo et al. [LCK*24] focuses on tracking whole body motions based on head mounted devices. Xu et al. [XSYP25] proposes a controller to reduce incorrect contacts or discontinuities for traversing new terrains. Tessler et al. [TGN*24] learns a physics-based controller to provide an intuitive control interface without requiring tedious reward engineering for all behaviors of interest. Juravsky et al. [JGFP24] trains controllers on thousands of diverse motion clips via progressive expert distillation. CLoSD [TRC*25] uses a diffusion model [CKCS26] for tracking. While existing tracking methods have advanced physics-based generation of single character, they cannot generalize into two-character interactions due to the force transfer and significantly increased complexity.

2.2. Physics-based Two-Character Interaction Synthesis

Although physics-based methods has shown promising results for individual characters performing a wide variety of behaviors, there exist only a few studies for multi-character animations. Park et al. [PRL*19] shows an example of chicken hopping and fighting with pre-defined discrete actions as well as target goals. While it shows the potential of physics-based two-character interaction synthesis, their interactions are simple and sparse. Won et al. [WGH21] presents a control strategy for two boxing or fencing characters by defining goal-oriented rewards. However, the control policy only demonstrates effectiveness on a specific interaction task with such task-specific rewards. Zhang et al. [ZGY*23] proposes a new reward formulation to facilitate various types of spatially and temporally dense interactions for full-body humanoid characters. However, their controllers are imitation controllers that cannot perform interactions that do not exist in the reference motions. Zhu et al. [ZZLH23] combines discrete latents with reinforcement learning to synthesize two-character boxing motions. However, their policy requires multiple stages to train and their control is specific

to tasks. Younes et al. [YKK*23] leverages adversarial imitation learning to generalize the idea of motion imitation for one character to deal with both the interaction for two physics-based characters. Their control policy is also specific to an interaction type and cannot generalize to multiple types. Liu et al. [LCDY24] focuses on reactive motions of one character when it interacts with the other character. Luo et al. [LWL*24] proposes a benchmark for simulating two-character interactions in the field of sports. However, their interactions are sparse and far from each other. Xie et al. [XSLvdP22] introduces a layer-wise mixture-of-experts architecture to integrate a diverse range of high-precision soccer juggling skills into a single physics-based character controller. Their system utilizes a task-description framework based on control graphs and success-adaptive random walks to facilitate the efficient learning of complex motor tasks and robust transitions between different body-part interactions. While kinematics-based interaction modeling [ZCMS25a; ZCMS25b; CWKS25] involve dense interactions, they do not consider physics. Our work focuses on generating contact-rich interactions within an all-in-one pipeline, which has been one of the challenging problems in physics-based two-character interaction animation.

3. Method

3.1. Motion Tracking

We aim to achieve physics-based motion tracking through training a policy π which enable a simulated humanoid character to produce a pose that closely resembles a kinematic target pose. The policy network is commonly conditioned on the future pose as a target for policy to imitate. At each timestep t , with observed state s_t and goal g_t , an RL agent interacts with an environment by applying an action a_t sampled from policy $a_t \sim \pi(a_t|s_t, g_t)$ and receives a reward r_t . The physics simulation environment defines transition dynamics $p(s_{t+1}|s_t, a_t)$ that produces the next state s_{t+1} . Similar to prior goal-conditioned reinforcement learning, we use the proximal policy gradient (PPO) to train the policy. The objective is to learn a policy that maximizes the discounted cumulative reward:

$$\mathbb{E}_{p(s_0)} \prod_{t=0}^{T-1} p(s_{t+1}|s_t, a_t) \pi(a_t|s_t, g_t) \left[\sum_{t=0}^T \gamma^t r_t \right], \quad (1)$$

where γ is the discount factor that reduces the weight of future rewards in PPO.

3.1.1. Observation

The observation input consists of humanoid state s_t and the goal state g_{t+1} that describes the target pose for the policy to imitate. Humanoid state $s_t = (s_t^p, s_t^v)$ contains local body positions s_t^p and linear velocities s_t^v .

Goal state $g_{t+1} = (\Delta s_{t+1}^p, \ominus s_{t+1}^q, \Delta s_{t+1}^v, \Delta s_{t+1}^a, s_{t+1}^{jp}, s_{t+1}^{jv})$ contains body positional offset Δs_{t+1}^p , rotational difference $\ominus s_{t+1}^q$, linear velocity offset Δs_{t+1}^v , angular velocity offset Δs_{t+1}^a , joint position s_{t+1}^{jp} and joint velocity s_{t+1}^{jv} .

3.1.2. Reward Function

The reward r_t encourages the agent to track the reference motion by minimizing the difference between the state of the simulated

character and the ground truth:

$$r_t^{\text{track}} = w^p r_t^p + w^q r_t^q + w^v r_t^v + w^a r_t^a + w^{jp} r_t^{jp} + w^{jv} r_t^{jv}, \quad (2)$$

where $w^{\{\cdot\}}$ denotes the respective weights, $r^{\{\cdot\}}$ denote the reward functions for tracking target body position, quaternion, linear velocity, angular velocity, joint position and joint velocity.

We also apply an energy penalty reward r_t^{energy} to minimize the high-frequency jitter. Total reward is calculated as:

$$r_t = r_t^{\text{track}} + 0.5 \times r_t^{\text{energy}}. \quad (3)$$

Detailed calculation and parameters of rewards can be found in supplementary material.

3.1.3. Action

Similar to prior work [PGH*22; LCKX*23; TGN*24], our policy generates the action $a_t \in \mathbb{R}^J$ which serves as the target for proportional derivative (PD) controller to apply torque at each joint. The action a_t is sampled from a multi-dimensional Gaussian distribution $a_t \sim \mathcal{N}(\bar{a}_t, \sigma)$, where \bar{a}_t is the mean action predicted by policy and $\sigma \in \mathbb{R}^J$ is learnable standard deviation.

3.1.4. Motion Sampling Strategy

To encourage sampling more challenging motions during policy training, we record the tracking rewards and adjust the sampling probabilities of different motion clips based on their recent tracking performance:

$$p_{m,t} = \text{Softmax}\left(\frac{-\bar{r}_m^{\text{track}}}{\mathcal{T}}\right), \quad (4)$$

where \mathcal{T} is the annealing temperature, \bar{r}_m^{track} is the recent average tracking reward of motion clip m and $p_{m,t}$ is the calculated sampling probability of motion clip m at timestep t .

3.2. Progressive Interaction Tracking

Training a single policy network for motion tracking on a large dataset could be difficult as it easily leads to catastrophic forgetting. Inspired by the continual learning paradigm of PNN, we introduce a model designed for tracking large-scale motion dataset automatically without hand-crafted dataset schedule of various difficulties.

3.2.1. Policy Model

Unlike vanilla PNNs that switch distinct experts for distinct tasks, our PNN experts operate additively. Each new expert does not replace the previous one but learns a residual action offset to correct the errors of the earlier frozen experts. The Gating Network is not a standard mixture-weight generator. Instead, it acts as a reward predictor that estimates how "confident" the experts are for a given state. The connection between them is governed by the routing ratio (Equation 7). Samples with low predicted rewards (low confidence) are routed to the new expert to learn the necessary corrective offsets.

In the context of this work, "difficulty" here specifically correlates to samples with stronger physical contact forces. A key contribution of our work is that we do not manually label difficulty. Instead, the system quantitatively defines "difficult" samples as those

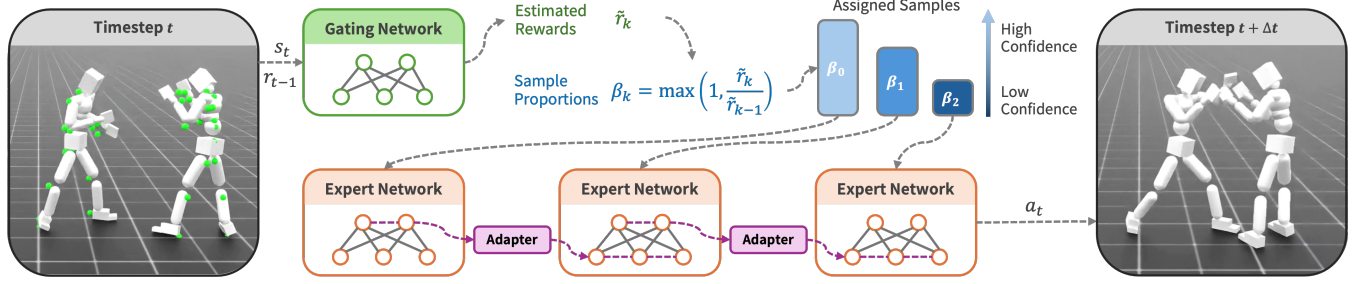


Figure 2: Framework overview. We train a progressive learning model in which later experts build on the knowledge from earlier experts, but specializing in more challenging motions. The policy receives humanoid state and goal state, and outputs actions for the proportional derivative (PD) controller to generate torques. Experts are activated sequentially, with adapters enabling knowledge transfer and a gating network estimating confidence.

with low estimated tracking rewards. These are the samples where previous experts fail to predict the correct joint torques required to maintain the pose against external perturbations.

Specifically, the entire policy model π contains k expert networks $\pi = (\pi_0, \dots, \pi_k)$ where each expert network is a 3-layer MLP with LeakyReLU activation. To facilitate efficient multitasking and prevent catastrophic forgetting, the architecture incorporates a gating network f_g and multiple lateral adapters [RRD*16] that connect sequential experts. These adapters function as knowledge transfer mechanisms, allowing newly activated experts to leverage the structural embeddings of previous frozen experts to learn residual action offsets. Instead of generating weights to blend expert outputs, the gating network is used for predicting the reward based on the input observation. Given an observation, gating network $f_g(s_t, g_t) : \mathbb{R}^d \rightarrow \mathbb{R}^k$ outputs an independent confidence of the experts:

$$\tilde{r}_k(s_t, g_t) = \text{Sigmoid}(f_g(s_t, g_t)_k), \quad (5)$$

where the Sigmoid activation maps the output to the unit interval $[0, 1]$, ensuring that the predicted confidence \tilde{r}_k is numerically consistent with the environment's normalized tracking rewards.

We start by training the first expert π_0 on the full dataset and all other experts are frozen. When the growth of estimated reward \tilde{r}_0 is stagnated, we stop updating the expert and activate a new one π_1 for learning harder motions by predicting the complementary offset actions on top of previous experts:

$$\pi_k(s_t, g_t) = \sum_{i < k} \pi_i(s_t, g_t). \quad (6)$$

However, it is difficult to seamlessly transition to a new expert. On one hand, the new expert needs to model the action distribution from scratch, which would lead to a significant drop in reward and time-consuming re-training. On the other hand, copying the parameters from previous expert to new one would inherit the bias and prevent expert from learning new knowledge.

We deploy two mechanisms to ensure stable transition from old to new expert while enabling the new one to learn novel knowledge effectively. First, we copy the parameters to all but the final layer, while zero-initializing the last layer and randomly initialize

the adapters connected to the last layer (which is different from the vanilla PNN). This is helpful to keep embedding capability inherited from old expert and ensure capacity for learning to adapt prior knowledge. Second, we propose a *Progressive Sampling Strategy* where the number of samples routed to the new expert depends on the estimated rewards compared to previous ones:

$$\beta_k = \max(1, \frac{\tilde{r}_k(s_t, g_t)}{\tilde{r}_{k-1}(s_t, g_t)}), \quad (7)$$

where β_k denotes the proportion of the samples with the lowest estimated rewards that are routed to expert π_k . This strategy encourages the new expert to prioritize learning harder motions. As the new expert improves, it is gradually exposed to samples in which the previous experts are more confident, until its performance saturates and a new expert is activated again.

3.2.2. Loss Function

During training, we activate only a single expert at a time and each expert k maintains its own learnable log standard deviation parameter $\log \sigma_k$. Using multiple $\log \sigma_k$ simultaneously in PPO causes unintended gradient updates in inactive experts. For the policy loss computation, we mask out the log standard deviations of all frozen experts, such that only the active expert's $\log \sigma_{k^*}$ is used when constructing the Gaussian distribution:

$$\pi_{k^*}(a_t | s_t) = \mathcal{N}(\bar{a}_t, \exp(2 \log \sigma_{k^*})). \quad (8)$$

This ensures that gradients flow exclusively through the active expert, while the log stds of inactive experts remain unchanged.

In order to encourage experts to reuse the learned knowledge through adapters, we also add an adapter usage loss:

$$u = \frac{\sum_{i \in \mathcal{A}_k} \|\theta_i^{\text{adapter}}\|}{\sum_{i \in \mathcal{A}_k} \|\theta_i^{\text{adapter}}\| + \|\theta_{k^*}^{\text{expert}}\|}, \quad (9)$$

$$\mathcal{L}_{\text{adapter}} = -\log(u + \epsilon), \quad (10)$$

where $\theta_{\{\cdot\}}$ is the weights of adapters or linear layer of expert, \mathcal{A}_k is the set of active adapters connecting to expert k and $\epsilon = 10^{-6}$.

For simplicity, we omit the detailed derivations of the policy and

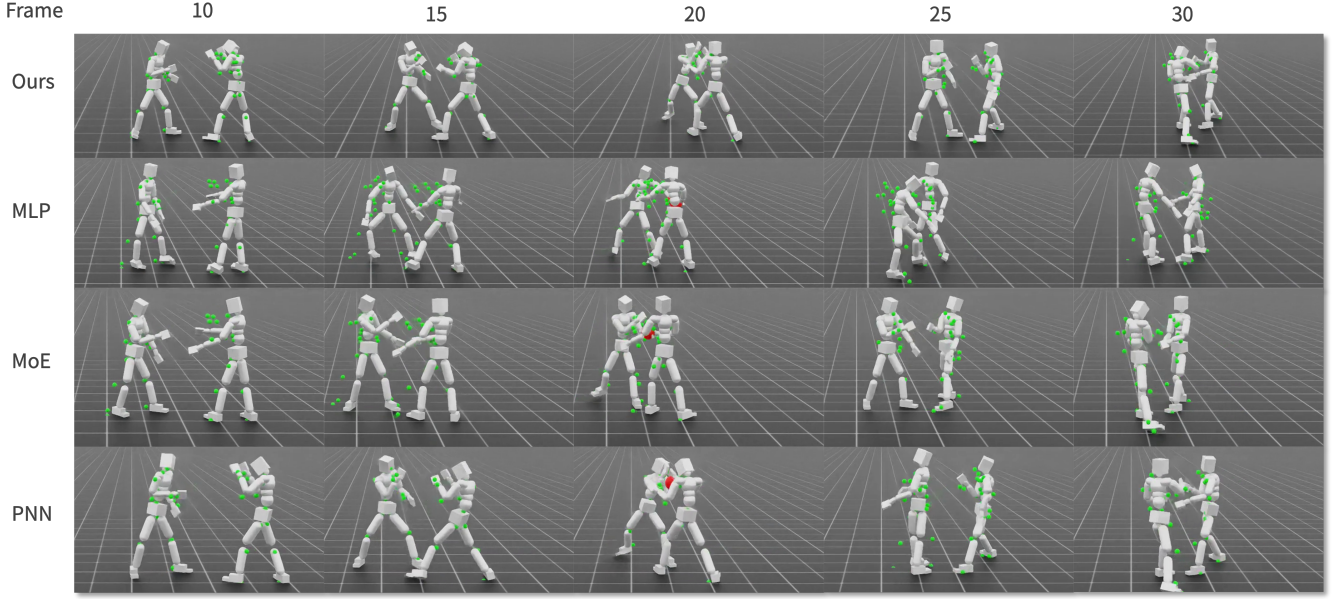


Figure 3: Qualitative comparison of tracking results across different models. From top to bottom, our method, MLP, MoE and PNN are shown that performs boxing interaction, respectively. Baseline models often exhibit instability or loss of balance under dense contact, while our method produces more stable and realistic interactions that closely follow the target motions.

value losses, as they follow the standard PPO formulation. Total loss function is formed as $\mathcal{L} = \mathcal{L}_{\text{policy}} + \mathcal{L}_{\text{value}} + 0.03 \times \mathcal{L}_{\text{adapter}}$.

For the gating network, it is updated independently from the tracking policy using the actual rewards from the environment:

$$\mathcal{L}_{\text{gating}} = \sqrt{\sum_k (r_k - \tilde{r}_k)^2}. \quad (11)$$

4. Experiments

4.1. Implementation Details

We conduct our experiments with an NVIDIA A100 GPU for training 4 experts. The physics simulation is run on NVIDIA Isaac Lab [MY*23]. We use SMPL [LMR*23] kinematic structure for the humanoid which contain 24 rigid bodies (pelvis as the root joint) and 69 degrees of freedom. Following prior work [PALV18], the starting frame of training episode is randomly selected from the sampled motion clip and the episode will be early terminated if the averaged rigid body distance to the target is lower than 0.5m. Our model is trained on InterHuman [LZL*24] which contains motion sequences of 1 million frames (10 hours at 30 fps).

4.2. Baselines

We compare our method against three baseline tracking policy models:

- A 3-layer multi-layer perceptron network (MLP) that represents the basic policy implementation such as SONIC [LYW*25].
- Mixture-of-experts network (MoE) which is representative for learning diverse skills used in previous works like DeepMimic [PALV18].

- Progressive Neural Network (PNN) [RRD*16] that represents the manual progressive strategies used in recent SOTA like PHC [LCKX*23]. PHC relies on a manually scheduled progressive network; our PNN baseline implements this exact manual strategy to demonstrate its limitations compared to our automatic routing. Specifically, the full dataset is partitioned into four subsets based on the averaged relative distances between the two humanoid robots, with later experts assigned to motions involving more frequent inter-body contacts.

4.3. Results

4.3.1. Tracking Performance

Following prior work [LCKX*23; TGN*24], in Table 1, we evaluate the tracking success rate (referred to as ‘Success’) as the ratio of successful episodes in which the average joint position error at every frame is less than 0.5m. We also report the mean per-joint position error (MPJPE) to assess the accuracy of alignment with the target pose. Despite the frequent perturbations of contact, our approach still achieves robust performance and a significantly higher success rate.

Figure 7 shows the tracking reward curves of all methods. MLP quickly saturates in the early stage and later degrades due to catastrophic forgetting. MoE achieves higher rewards than MLP with its larger capacity but still suffers from forgetting. PNN exhibits sharp drops when new experts are activated, caused by the distribution shift across motion subsets, and also incurs higher training cost since each expert must be trained from scratch. In contrast, our method exhibits substantially more stable transitions when introducing new experts and requires considerably less training time.

Moreover, it automatically routes appropriate number of samples to the active experts, thereby obviating the need for manual dataset scheduling across experts in progressive learning.

As illustrated in Figure 6, our method also demonstrates strong adaptability when the interaction skill transitions abruptly from spinning to boxing. Despite the sudden change in motion dynamics, the characters maintain stable coordination without collapsing into unnatural states. This indicates that the progressive expert routing effectively preserves prior knowledge while enabling quick adaptation to new interaction modes. Notably, the system avoids discontinuous prediction across multiple experts, which are common in skill-specific controllers such as PNN.

	Train Set		Test Set	
	Success \uparrow	MPJPE \downarrow	Success \uparrow	MPJPE \downarrow
MLP	53.3%	81.3	28.5%	117.0
MoE	76.2%	55.7	64.0%	69.4
PNN	82.3%	50.9	78.7%	58.1
Ours	91.8%	37.9	85.2%	43.4

Table 1: Comparison of tracking performance across baselines and our method on InterHuman dataset. We report success rate and mean per-joint position error (MPJPE) on both training and test sets.

We further evaluate our method on a subset of the AMASS dataset to examine its performance in single-character tracking scenarios. This subset consists of approximately 200 motion clips, totaling 300 minutes of motion sequences at 30 fps. For the PNN baseline, the dataset is partitioned into four subsets assigned to four corresponding experts based on averaged joint velocities. As shown in Table 2, all models demonstrate higher tracking success rates on AMASS compared to the InterHuman dataset, which is primarily attributed to the absence of inter-body perturbations inherent in two-character interactions.

	Train Set		Test Set	
	Success \uparrow	MPJPE \downarrow	Success \uparrow	MPJPE \downarrow
MLP	71.5%	56.0	66.5%	72.3
MoE	86.3%	47.2	82.9%	53.1
PNN	91.6%	42.8	88.4%	46.5
Ours	96.7%	36.1	93.0%	40.2

Table 2: Comparison of tracking performance across baselines and our method on AMASS dataset.

4.3.2. Perturbations

To assess robustness, we introduce perturbations to the tracked humanoid by randomly throwing objects of varying masses and by injecting noise into the input observations. In these experiments, all models, including the baselines, are retrained in environments where such perturbations are applied during training. Quantitative and qualitative results are reported in Table 3.

When tested with external disturbances, our approach sustains

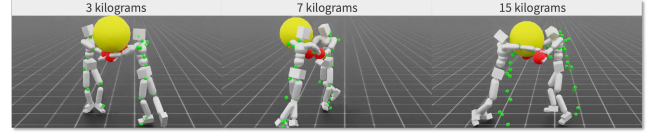


Figure 4: Tracking under external perturbations with different object masses (3 kg, 7 kg, 15 kg). As the perturbation strength increases, the characters experience growing difficulty in maintaining stable interaction.

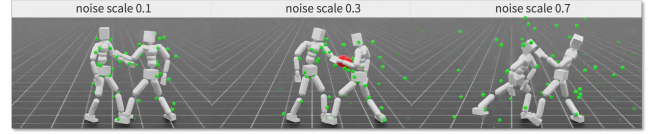


Figure 5: Tracking under observation noise with different noise scales (0.1, 0.3, 0.7). Larger noise levels lead to instability and loss of balance in the interactions.

high success rates, while baseline methods show marked degradation. Notably, PNN fails under perturbations due to its gating network's reliance on dataset-specific specialization, which does not generalize when noise shifts the input distribution. MoE handles perturbations slightly better through soft blending, but still lacks sufficient adaptability. Since MoE and MLP are already prone to catastrophic forgetting, they perform even worse in noisy environments compared to safe ones. Our experts, trained progressively with sample routing, exhibit significantly stronger resilience to observation noise and external force perturbations. This suggests that expert specialization in our framework is not brittle but rather complementary, where later experts refine challenging behaviors without overwriting earlier skills.

	Object Perturb		Noise Inject	
	Success \uparrow	MPJPE \downarrow	Success \uparrow	MPJPE \downarrow
MLP	14.5%	177.6	20.3%	139.9
MoE	52.4%	90.6	57.1%	81.1
PNN	55.3%	84.3	59.5%	77.0
Ours	75.3%	59.0	80.2%	52.9

Table 3: Robustness evaluation under perturbations. Performance is measured when randomly throwing objects of varying masses (Object Perturb) and injecting noise into input observations (Noise Inject).

4.3.3. Expert Contribution

In our method, later experts enhance learning capacity by specializing in more challenging input samples and predicting action offsets for earlier experts. The results in Table 4 sheds light on how different experts contribute during training. We observe that each newly activated expert reduces both MPJPE and training time, which indicates that later experts act as refiners by predicting action offsets relative to earlier experts. Interestingly, the marginal improvement between the fourth and fifth expert is small, suggesting diminishing

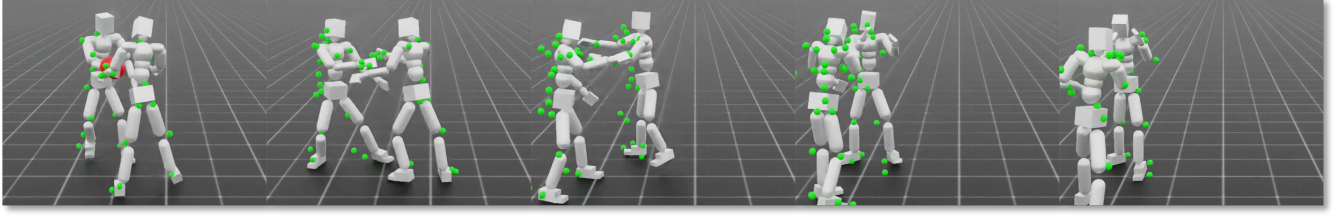


Figure 6: Tracking with interaction skill transitions. The target interaction shifts from spinning to boxing, and our system remains robust to these abrupt changes without failure.

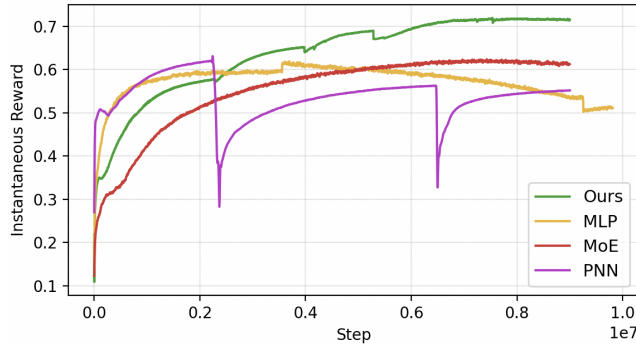


Figure 7: Training reward curves of baselines and our method. Our approach shows smooth transitions between experts and faster convergence compared to PNN.

returns once the model capacity surpasses the scale of InterHuman. This analysis confirms that four experts are sufficient for the present dataset, and that our automatic routing strategy naturally balances capacity and efficiency without requiring manual dataset scheduling more quickly.

Another important observation is that our approach substantially reduces training time compared to PNN. In PNN, each expert is trained nearly from scratch on a subset of data, resulting in redundant training and sharp reward drops. Our method mitigates this by transferring knowledge across experts, as shown by the smooth reward curve (Figure 7). This not only improves efficiency but also avoids instability during expert activation, which is crucial for scaling to larger datasets.

	Success \uparrow	MPJPE \downarrow	Training Time (hours)
Expert 1	61.1%	83.4	29
Expert 2	72.6%	60.2	19
Expert 3	82.7%	47.5	15
Expert 4	85.2%	43.4	7
Expert 5	85.8%	42.7	2

Table 4: Contribution of different experts in our progressive framework. Later experts improve success rate and MPJPE while requiring less training time, showing that specialization accelerates learning.

4.3.4. Ablation Study

We conduct experiments ablating key components of our framework: (1) Progressive Sampling Strategy (PSS), where all new experts are trained on the full dataset when initiated instead of routing based on estimated rewards; (2) adapters, removing the lateral connections between experts that enable knowledge transfer; and (3) adapter loss, omitting the regularization term that encourages adapter usage (Equation 10).

	Success \uparrow	MPJPE \downarrow	Training Time (hours)
Full Setup	85.2%	43.4	70
w/o PSS	68.0%	64.1	167
w/o adapter	76.3%	60.9	122
w/o adapter loss	82.3%	49.1	70

Table 5: Qualitative results of full setup and ablated versions of our method.

Removing Progressive Sampling Strategy causes the largest performance drop and substantially increases training time. This reveals that the model wastes capacity re-learning easy samples for each expert instead of specializing on hard, contact-heavy failures. Ablating the adapters also degrades tracking quality and slows training. This shows that progressive experts cannot function in isolation. Without lateral knowledge transfer, each new expert is forced to re-learn large parts of the representation space, leading to redundant computation. Removing the adapter loss leads to a smaller but noticeable decline in performance. The loss acts mainly as a fine-tuning mechanism rather than a structural necessity. It gently encourages later experts to reuse previous knowledge via adapters instead of over-specializing or ignoring them.

4.3.5. Motion Sampling Strategy

We further evaluate the impact of different motion sampling strategies on performance. Specifically, we compare four approaches: (1) *Uniform*, where all motion clips are sampled with equal probability; (2) *Motion Duration*, where longer clips are assigned higher sampling probabilities; (3) *Success Rate*, where clips with lower success rates are prioritized to encourage training on more difficult motions; and (4) *Tracking Reward*, where clips with lower tracking rewards are sampled more frequently to emphasize challenging cases. In Table 6, we report both the success rate and the mean episode length (referred to as ‘Episode’) in training stage. The mean episode length reflects the average duration of simulation an agent survives before the episode ends, either due to task

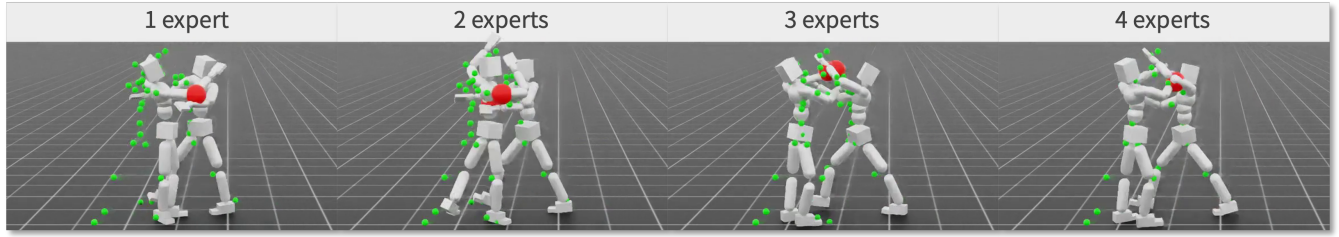


Figure 8: Ablation on the number of experts in our progressive framework. With only 1–2 experts, the characters often fail to maintain stable interactions. Adding more experts (3–4) improves tracking quality, showing that later experts specialize in handling more challenging motion dynamics.

termination (e.g., falling or large tracking error) or truncation at the maximum allowed horizon.

Table 6 highlights how different motion sampling strategies influence tracking performance. The Uniform strategy yields the lowest success rate and episode length, as it does not differentiate between motions of varying complexity. Motion Duration provides a modest improvement, suggesting that longer clips contain more diverse patterns that benefit learning. However, the gains remain limited because this strategy does not explicitly prioritize difficult motions.

In contrast, Success Rate sampling substantially improves both success and episode length. By down-weighting easier motions and emphasizing those with lower success, the policy is exposed to more challenging interactions. Similarly, Tracking Reward achieves the best overall performance. Prioritizing motions with lower tracking reward ensures that the policy repeatedly trains on failure-prone sequences, which leads to longer survival time during episodes. Interestingly, the difference between Success Rate and Tracking Reward is relatively small, but the latter consistently achieves the highest values. This suggests that reward-based sampling provides a finer-grained signal of difficulty compared to binary success or failure.

	Success [†]	Episode (seconds) [†]
Uniform	74.4%	14.9
Motion Duration	81.7%	17.1
Success Rate	88.4%	20.5
Tracking Reward	91.8%	21.2

Table 6: Impact of motion sampling strategies. Adaptive sampling based on success rate or tracking reward leads to higher success and longer average episode length compared to uniform or duration-based sampling.

5. Conclusion

We introduced a progressive mixture-of-experts framework for physics-based motion tracking of contact-rich interactions. By progressively expanding expert capacity and automatically routing training samples, our method achieves stable and efficient learning without requiring manual dataset scheduling. Experiments on large-scale datasets demonstrate that our approach surpasses prior

baselines in tracking accuracy, robustness under perturbations, and training efficiency. Ablation studies further confirm that later experts specialize in more challenging dynamics while maintaining smooth knowledge transfer across the model. We believe this all-in-one progressive framework offers an extensible foundation for the community to advance research in motion tracking without reliance on task-specific controllers or fragmented training pipelines.

References

- [CKCS26] CHANG, ZIYI, KOULIERIS, GEORGE A, CHANG, HYUNG JIN, and SHUM, HUBERT PH. “On the design fundamentals of diffusion models: A survey”. *Pattern Recognition* 169 (2026), 111934 [2](#).
- [CMM*18] CHENTANEZ, NUTTAPONG, MÜLLER, MATTHIAS, MACKLIN, MILES, et al. “Physics-based motion capture imitation with deep reinforcement learning”. *Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games*. 2018, 1–10 [2](#).
- [CWKS25] CHANG, ZIYI, WANG, HE, KOULIERIS, GEORGE, and SHUM, HUBERT P. H. “Large-Scale Multi-Character Interaction Synthesis”. *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*. SIGGRAPH Conference Papers '25. Association for Computing Machinery, 2025. ISBN: 9798400715402. DOI: [10.1145/3721238.3730750](https://doi.org/10.1145/3721238.3730750). URL: <https://doi.org/10.1145/3721238.3730750>.
- [FBH21] FUSSELL, LEVI, BERGAMIN, KEVIN, and HOLDEN, DANIEL. “Supertrack: Motion tracking for physically simulated characters using supervised learning”. *ACM Transactions on Graphics (TOG)* 40.6 (2021), 1–13 [2](#).
- [JGFP24] JURAVSKY, JORDAN, GUO, YUNRONG, FIDLER, SANJA, and PENG, XUE BIN. “Superpadl: Scaling language-directed physics-based control with progressive supervised distillation”. *ACM SIGGRAPH 2024 Conference Papers*. 2024, 1–11 [2](#).
- [LCC*24] LUO, ZHENGYI, CAO, JINKUN, CHRISTEN, SAMMY, et al. “OmniGrasp: Grasping diverse objects with simulated humanoids”. *Advances in Neural Information Processing Systems* 37 (2024), 2161–2184 [2](#).
- [LCDY24] LIU, YUNZE, CHEN, CHANGXI, DING, CHENJING, and YI, LI. “PhysReaction: Physically plausible real-time humanoid reaction synthesis via forward dynamics guided 4d imitation”. *Proceedings of the 32nd ACM International Conference on Multimedia*. 2024, 3771–3780 [3](#).
- [LCK*24] LUO, ZHENGYI, CAO, JINKUN, KHIRODKAR, RAWAL, et al. “Real-time simulated avatar from head-mounted sensors”. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, 571–581 [2](#).
- [LCKX*23] LUO, ZHENGYI, CAO, JINKUN, KITANI, KRIS, XU, WEIPENG, et al. “Perpetual humanoid control for real-time simulated avatars”. *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, 10895–10904 [1–3](#), [5](#).

- [LCM*23] LUO, ZHENGYI, CAO, JINKUN, MEREL, JOSH, et al. “Universal humanoid motion representations for physics-based control”. *arXiv preprint arXiv:2310.04582* (2023) 2.
- [LMR*23] LOPER, MATTHEW, MAHMOOD, NAUREEN, ROMERO, JAVIER, et al. “SMPL: A skinned multi-person linear model”. *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 2023, 851–866 5.
- [LWL*24] LUO, ZHENGYI, WANG, JIASHUN, LIU, KANGNI, et al. “Smplolympics: Sports environments for physically simulated humanoids”. *arXiv preprint arXiv:2407.00187* (2024) 1–3.
- [LYW*25] LUO, ZHENGYI, YUAN, YE, WANG, TINGWU, et al. “Sonic: Supersizing motion tracking for natural humanoid whole-body control”. *arXiv preprint arXiv:2511.07820* (2025) 5.
- [LZL*24] LIANG, HAN, ZHANG, WENQIAN, LI, WENXUAN, et al. “Intergen: Diffusion-based multi-human motion generation under complex interactions”. *International Journal of Computer Vision* (2024), 1–21 2, 5.
- [MYY*23] MITTAL, MAYANK, YU, CALVIN, YU, QINXI, et al. “Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments”. *IEEE Robotics and Automation Letters* 8.6 (2023), 3740–3747. DOI: 10.1109/LRA.2023.3270034 5.
- [PALV18] PENG, XUE BIN, ABBEEL, PIETER, LEVINE, SERGEY, and VAN DE PANNE, MICHIEL. “Deepmimic: Example-guided deep reinforcement learning of physics-based character skills”. *ACM Transactions On Graphics (TOG)* 37.4 (2018), 1–14 1, 2, 5.
- [PGH*22] PENG, XUE BIN, GUO, YUNRONG, HALPER, LINA, et al. “Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters”. *ACM Transactions On Graphics (TOG)* 41.4 (2022), 1–17 3.
- [PRL*19] PARK, SOOHWAN, RYU, HOSEOK, LEE, SEYOUNG, et al. “Learning predict-and-simulate policies for unorganized human motion data”. *ACM Transactions on Graphics (TOG)* 38.6 (2019), 1–11 2.
- [RRD*16] RUSU, ANDREI A, RABINOWITZ, NEIL C, DESJARDINS, GUILLAUME, et al. “Progressive Neural Networks”. (2016) 2, 4, 5.
- [RYC*23] REN, JIAWEI, YU, CUNJUN, CHEN, SIWEI, et al. “Diffmimic: Efficient motion mimicking with differentiable physics”. *arXiv preprint arXiv:2304.03274* (2023) 2.
- [TGN*24] TESSLER, CHEN, GUO, YUNRONG, NABATI, OFIR, et al. “Maskedmimic: Unified physics-based character control through masked motion inpainting”. *ACM Transactions on Graphics (TOG)* 43.6 (2024), 1–21 2, 3, 5.
- [TRC*25] TEVET, GUY, RAAB, SIGAL, COHAN, SETAREH, et al. “CLoSD: Closing the Loop between Simulation and Diffusion for multi-task character control”. *The Thirteenth International Conference on Learning Representations*. 2025 2.
- [WGH20] WON, JUNGDM, GOPINATH, DEEPAK, and HODGINS, JESSICA. “A scalable approach to control diverse behaviors for physically simulated characters”. *ACM Transactions on Graphics (TOG)* 39.4 (2020), 33–1 2.
- [WGH21] WON, JUNGDM, GOPINATH, DEEPAK, and HODGINS, JESSICA. “Control strategies for physically simulated characters performing two-player competitive sports”. *ACM Transactions on Graphics (TOG)* 40.4 (2021), 1–11 1, 2.
- [WGSF20] WANG, TINGWU, GUO, YUNRONG, SHUGRINA, MARIA, and FIDLER, SANJA. “Unicon: Universal neural controller for physics-based character motion”. *arXiv preprint arXiv:2011.15119* (2020) 2.
- [XSLvdP22] XIE, ZHAOMING, STARKE, SEBASTIAN, LING, HUNG YU, and van de PANNE, MICHIEL. “Learning Soccer Juggling Skills with Layer-wise Mixture-of-Experts.(2022)”. (2022) 3.
- [XSYP25] XU, MICHAEL, SHI, YI, YIN, KANGKANG, and PENG, XUE BIN. “Parc: Physics-based augmentation with reinforcement learning for character controllers”. *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Papers*. 2025, 1–11 2.
- [YKK*23] YUAN, YE and KITANI, KRIS. “Residual force control for agile human behavior imitation and extended motion synthesis”. *Advances in Neural Information Processing Systems* 33 (2020), 21763–21774 2.
- [YOUNES*23] YOUNES, MOHAMED, KIJAK, EWA, KULPA, RICHARD, et al. “MAAIP: Multi-Agent Adversarial Interaction Priors for imitation from fighting demonstrations for physics-based characters”. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 6.3 (2023), 1–20 3.
- [ZCMS25a] ZHANG, XIAOTANG, CHANG, ZIYI, MEN, QIANHUI, and SHUM, HUBERT P. H. “Motion In-Betweening for Densely Interacting Characters”. *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*. SA Conference Papers '25. Association for Computing Machinery, 2025. ISBN: 9798400721373. DOI: 10.1145/3757377.3763950. URL: <https://doi.org/10.1145/3757377.3763950> 3.
- [ZCMS25b] ZHANG, XIAOTANG, CHANG, ZIYI, MEN, QIANHUI, and SHUM, HUBERT PH. “Real-Time and Controllable Reactive Motion Synthesis via Intention Guidance”. *Computer Graphics Forum*. Vol. 44. 6. Wiley Online Library. 2025, e70222 3.
- [ZGY*23] ZHANG, YUNBO, GOPINATH, DEEPAK, YE, YUTING, et al. “Simulation and retargeting of complex multi-character interactions”. *ACM SIGGRAPH 2023 Conference Proceedings*. 2023, 1–11 2.
- [ZZLH23] ZHU, QINGXU, ZHANG, HE, LAN, MENGTING, and HAN, LEI. “Neural categorical priors for physics-based character control”. *ACM Transactions on Graphics (TOG)* 42.6 (2023), 1–16 1, 2.