# *Supplementary Material* for Humanoid Generative Pre-Training for Zero-Shot Motion Tracking

## 1. Additional Visualization

In this section, we provide additional real-robot examples, as shown in Fig. 1, including both teleoperation demonstrations and zero-shot dancing. As a powerful zero-shot tracker, Humanoid-GPT can execute a wide range of complex behaviors, such as playing basketball, collaboratively carrying boxes with a human partner, and even rolling over and standing up from the ground. We also showcase more iconic dance routines, where motions are directly captured from videos and retargeted to the G1 space; these sequences are not included in our training set.

## 2. Additional Ablation Studies

### 2.1. Number of Experts and Cluster Granularity

We next study the effect of the number of motion experts and the granularity of clusters produced by the Harmonic Motion Embedding (HME) representation. We vary the number of clusters $C \in \{128, 256, 384, 512, 1024\}$ while keeping the total training corpus fixed, leading to different numbers of experts. For each configuration, we train a distilled Humanoid-GPT-B model on the corresponding expert set and evaluate on the AMASS test split.

As indicated in Fig. 2, extremely coarse clustering (e.g., 128 experts) leads to experts that cover overly heterogeneous motion patterns, which harms teacher tracking fidelity. Overly fine granularity (e.g., 1024 experts) increases training cost with conflicting guidelines for the students. The configuration with roughly $C \approx 384$ experts offers the best balance between diversity, per-cluster coherence, and compute.

### 2.2. History Length of Transformer

Compared with MLPs, Transformers not only offer greater scalability, but more importantly provide substantially enhanced temporal modeling. MLP-based policies typically condition on only a single historical frame, whereas Transformers are inherently designed for sequence modeling. In Fig. 2, we present the effect of varying sequence length on model performance. All experiments use a Base-sized model with default hyperparameters under a controlled single-factor setting. As shown in the figure, performance continues to improve even with a history of 64 frames. However, due to the quadratic increase in computation with sequence length, we adopt 32 historical frames as the default setting.

### 2.3. Environment Number for DAgger Rollout

As we scaled up the training data, we found that the number of environments in DAgger also needed to increase accordingly, as shown in Fig. 2. We ultimately adopted 32K environments. We hypothesize that this is due to the large number of reference motions, where using too few environments may lead to overfitting and forgetting.

## 3. Implementation and Reproducibility Details

This section expands the method details that could not fit in the main paper. We include RL hyperparameters, DAgger schedules, and full compute accounting.

### 3.1. RL Expert Training Details

**Environment and episode setup.** For each skill cluster, we train a PPO expert in MuJoCo using randomized episode lengths ranging from $T_{\min}$ to $T_{\max}$ frames (typically 600–1200). The control loop runs at 50 Hz, and the reference motion is downsampled to match this frequency. Episodes terminate upon detecting a fall, encountering excessive joint-limit violations, or reaching the time-out horizon. All hyperparameters and reward weights involved in PPO training are listed in Table 4 and Table 3.

To enhance robustness and improve generalization across diverse motion clusters, we apply a set of domain randomization strategies during training. These perturbations are injected into the MuJoCo environment at the beginning of each episode and occasionally throughout rollout generation, ensuring that the learned policy remains stable under variations in dynamics, sensing, and execution conditions. As shown in Table 1, we randomize:

- **Terrain properties**, including floor friction, maximum terrain height, and procedural noise parameters for terrain generation (noise scale, octaves, persistence, and lacunarity).
- **External forces**, where both the interval between force

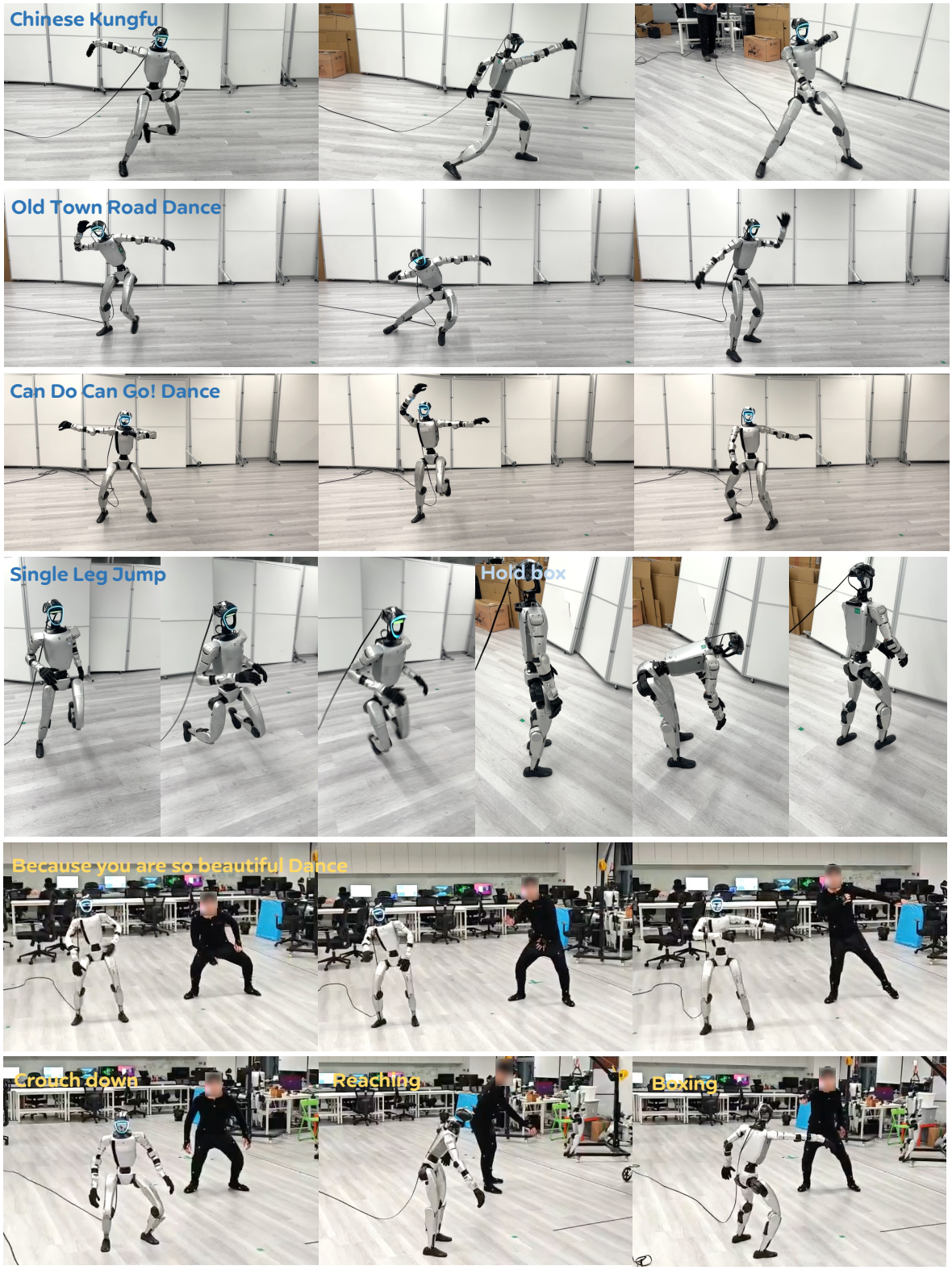


Figure 1. **Additional Real-world experiments for our Humanoid-GPT.** All motions illustrated are excluded from training to verify generalization capability. Our method can track diverse, complex and high-dynamic motion in a zero-shot manner, especially various dance motions.

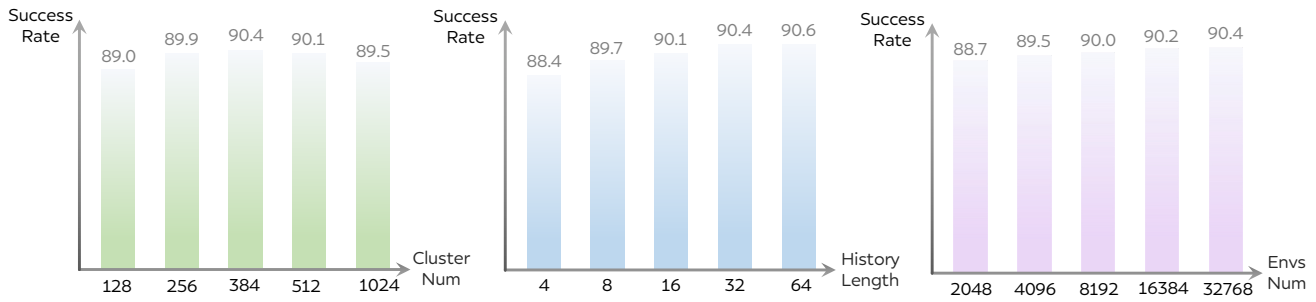


Figure 2. **Ablation studies for Humanoid-GPT.**

Table 1. **Domain Randomizations.**

| Item | Random range |
|---|---|
| **Terrains** | |
| Floor friction | $\mathcal{U}(0.3,\ 2.0)$ |
| Max terrain height | $0.3$ |
| Noise scale | $\mathcal{U}(10.0,\ 16.0)$ |
| Noise octaves | $\mathcal{U}(5.0,\ 8.0)$ |
| Noise persistence | $\mathcal{U}(0.3,\ 0.5)$ |
| Noise lacunarity | $\mathcal{U}(2.0,\ 4.0)$ |
| **External Forces** | |
| Interval range | $\mathcal{U}(5.0,\ 10.0)$ |
| Velocity magnitude range | $\mathcal{U}(0.1,\ 1.0)$ |
| **Physical Property Changes** | |
| DoF friction scaling | $\mathcal{U}(0.5,\ 2.0)$ |
| Armature scaling | $\mathcal{U}(1.0,\ 1.05)$ |
| Torso CoM position change | $\mathcal{U}(-0.15,\ 0.15)$ |
| Torso mass change | $\mathcal{U}(-3.0,\ 6.0)$ |
| Default DoF position jittering | $\mathcal{U}(-0.05,\ 0.05)$ |

injections and the magnitude of the applied velocity perturbations are sampled from uniform distributions.

- **Physical property variations**, including DoF friction scaling, armature scaling, torso center-of-mass shifts, torso mass perturbations, and per-DoF position jittering at the beginning of each rollout.

These domain randomization settings are applied uniformly across all PPO expert training environments. For the DAgger stage, we use the same environment configuration and identical randomization scheme, ensuring that the aggregated demonstrations capture the full distribution of dynamic variations encountered by the expert policy.

### 3.2. DAgger Distillation Schedule

We follow a standard DAgger loop for Behaviour Cloning (BC):

1. Initialize both the expert teacher and the student policy within the simulation environment.
2. At iteration $i$, roll out the student policy and query

Table 2. **Hyperparameter settings for training motion experts.**

| Hyperparameter | Value |
|---|---|
| Env Numbers | 32768 |
| Batch size | 1024 |
| Discount factor $\gamma$ | 0.97 |
| GAE parameter $\lambda$ | 0.95 |
| Clipping parameter $\epsilon$ | 0.2 |
| Policy network size | [512, 256, 128] |
| Critic network size | [512, 256, 128] |
| Learning rate | $3 \times 10^{-4}$ |
| Entropy coefficient | 0.01 |
| Optimizer | Adam |
| Training iteration per expert | 3B |

Table 3. **Reward weights of different terms.**

| Term | Weight |
|---|---|
| lowerbody keypoints $w_k$ | 1.5 |
| upperbody keypoints $w_k$ | 0.75 |
| keypoint position $\alpha_{\text{pos}}$ | 1.0 |
| keypoint orientation $\alpha_{\text{rot}}$ | 2.0 |
| keypoint linear velocity $\alpha_{\text{vel}}$ | 0.03 |

Table 4. **Hyperparameter settings for DAgger BC.**

| Hyperparameter | Value |
|---|---|
| Env Numbers | 32768 |
| Batch size | 32768 |
| Gradient Clipping | 1.0 |
| Learning rate | $1 \times 10^{-4}$ |
| Num Layers | 12 |
| Channel dims | 256/384/768 |
| Optimizer | AdamW |
| Training iteration | 200k |

the expert for the corresponding target action using the same state.

3. Train the student to match the expert's action, and then update the environment using the student's executed action.

In practice, we fix the maximal history length $H$ in Eq. (2) to 32 and maintain history buffers for both the teacher's actions and the student's observations. To avoid mode collapse when some experts cover only partial behavior distributions, the batch size used for Behaviour Cloning is kept no smaller than the number of experts.

### 3.3. Compute Cost Breakdown

The main paper reports a total compute budget of roughly 15,000 GPU hours. Here we detail the breakdown between expert training and Transformer distillation shown in Table 5.

We emphasize that, once trained, only the distilled Humanoid-GPT policy is required at deployment time. The expert library is used solely as a training-time teacher and can be discarded afterwards.

### 3.4. Deployment Details and Latency Measurements

For completeness, we provide the exact configuration used to obtain the latency results in Fig. 5 of the main paper:

- **Inference hardware:** Single NVIDIA RTX 4090 GPU, CPU: *Intel Core i9-14900KF*.
- **ONNX export:** FP32 weights with CUDA.
- **TensorRT optimization:** Engine built with optimized kernels for causal attention and fused MLPs.
- **Control loop:** End-to-end closed-loop frequency of 50 Hz, including sensor read, inference, PD computation, and actuation commands.

The complete deployment stack will be released as configuration files and scripts, enabling reproducible real-time control on G1-like humanoids.

## 4. Scaling Laws

The main paper qualitatively demonstrates monotonic scaling trends. Here we provide a more formal analysis of scaling laws and quantify the relationship between dataset diversity and generalization.

### 4.1. Data Scaling Up

We vary the number of training tokens $T \in \{2\text{M}, 20\text{M}, 200\text{M}, 2\text{B}\}$ using the Humanoid-GPT-B architecture. For each $T$, we sample a subset of the 2B-frame corpus without overlap across subsets. As shown in Fig. 3. We also observe that the marginal gains decrease slightly between 200M and 2B tokens, suggesting the onset of a data-limited regime for the current model capacity.

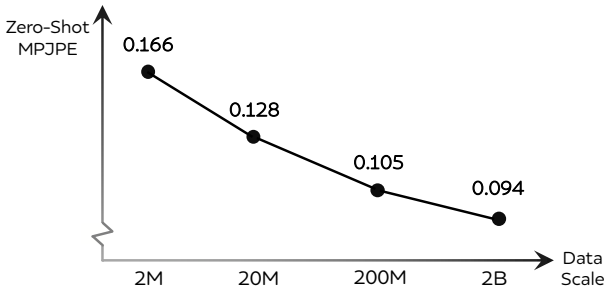


Figure 3. **Data Scaling up Curve on Zero-shot Performance.**

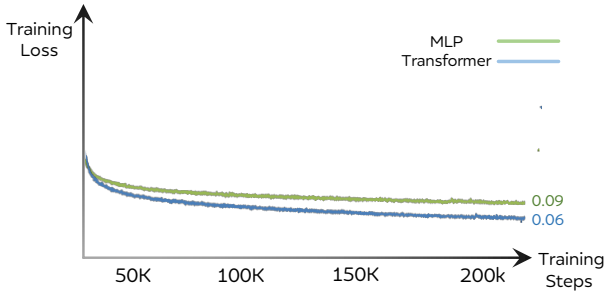


Figure 4. **Model Scalability Comparison.**

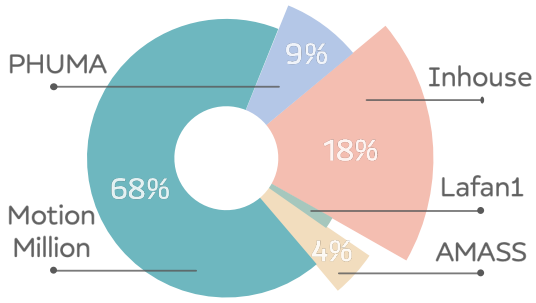


Figure 5. **Data distribution visualization.**

### 4.2. Model Scaling Ability

We evaluate the scalability of our model by comparing a Transformer-B architecture with an MLP of comparable parameter size, both trained on 2B tokens, with results summarized in Fig. 4. We observe that the Transformer continues to improve steadily as training progresses, whereas the MLP saturates early, which demonstrates the scalability of Humanoid-GPT.

### 4.3. Compared methods.

We compare the following baseline policies, which represent the strongest publicly available humanoid trackers at the time of writing and are all based on MLP-style low-level controllers trained on around 6–9M motion frames:

Table 5. **Approximate compute breakdown.**

| Stage | Hardware | Total GPU hours | Fraction of total (%) |
|---|---|---|---|
| PPO experts ($\sim$ 384 experts) | RTX 4090 | 12,000 | 75% |
| Distillation (Humanoid-GPT-S/B/L) | H100 | 3,000 | 25% |
| **Total** | — | **15,000** | **100%** |

- **GMT** [1]: A Mixture-of-Experts (MoE) tracker trained on a subset of AMASS [3] motions, where each expert specializes in a particular motion pattern and the gating network selects appropriate experts to maintain physically consistent whole-body tracking.
- **TWIST** [4]: A whole-body imitation policy distilled from the TWIST teleoperation system, designed for responsive human-in-the-loop control on Unitree humanoids and trained on a large corpus of teleoperated demonstrations covering everyday and dynamic behaviors.
- **Any2Track** [5]: A general motion tracker trained on AMASS [3] and LAFAN1 [2], which emphasizes robustness to perturbations by incorporating dynamics-adaptive control objectives and strong disturbance randomization during training.

For all three methods, we use the authors' released implementations and checkpoints and evaluate them under the same simulation and retargeting protocol as our Humanoid-GPT models to ensure a fair comparison.

## References


[1] Zixuan Chen, Mazeyu Ji, Xuxin Cheng, Xuanbin Peng, Xue Bin Peng, and Xiaolong Wang. Gmt: General motion tracking for humanoid whole-body control. *arXiv preprint arXiv:2506.14770*, 2025. 5

[2] Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)*, 39(4):60–1, 2020. 5

[3] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019. 5

[4] Yanjie Ze, Zixuan Chen, JoÃĢo Pedro AraÃšjo, Zi-ang Cao, Xue Bin Peng, Jiajun Wu, and C Karen Liu. Twist: Teleoperated whole-body imitation system. *arXiv preprint arXiv:2505.02833*, 2025. 5

[5] Zhikai Zhang, Jun Guo, Chao Chen, Jilong Wang, Chenghuai Lin, Yunrui Lian, Han Xue, Zhenrong Wang, Maoqi Liu, Jiangran Lyu, et al. Track any motions under any disturbances. *arXiv preprint arXiv:2509.13833*, 2025. 5