Prime and Reach: Synthesising Body Motion for Gaze-Primed Object Reach

Supplementary Material

Contents

A. Qualitative Videos

We include the qualitative video on our website `https://masashi-hatano.github.io/prime-and-reach/` showcasing predicted motion sequences from our P&R model over different datasets. For each of the sequences, we provide the goal location in green sphere, our goal-pose conditioned prediction in yellow, and the goal-location conditioned synthesis in brown.

B. Further Details on P&R sequence curation

B.1. Slab Test Method for Priming

The Slab Test Method expects the knowledge of the target location o_{3D}, which is an axis-aligned 3D bounding box, defined by its minimum ($\mathbf{b}_{\text{min}}$) and maximum ($\mathbf{b}_{\text{max}}$) corners, or as 3D coordinates of the object center.

The Slab Test Method treats the box as the overlapping volume of three infinite slabs (one for each axis), each bounded by a pair of parallel planes. A visualisation of the intersection checks is shown in Figure S1. The algorithm calculates two key parametric distances along the gaze ray. The first, t_{near}, represents the distance to the last slab plane that the ray enters. It is the furthest entry point, marking the moment the ray is inside all three slabs and thus inside the box. The second, t_{far}, is the distance to the first slab plane that the ray exits. It is the nearest exit point, marking the moment the ray leaves the box volume. A valid intersection occurs if the ray enters the box before it exits, as defined by the condition in Equation 1,

$$t_{\text{near}} = \max_{i \in x,y,z} \min \left(\frac{\mathbf{b}_{\text{min}}^{(i)} - \mathbf{o}_{\text{cam}}^{(i)}}{\hat{\mathbf{d}}_{\text{gaze}}^{(i)}}, \frac{\mathbf{b}_{\text{max}}^{(i)} - \mathbf{o}_{\text{cam}}^{(i)}}{\hat{\mathbf{d}}_{\text{gaze}}^{(i)}} \right)$$
$$t_{\text{far}} = \min_{i \in x,y,z} \max \left(\frac{\mathbf{b}_{\text{min}}^{(i)} - \mathbf{o}_{\text{cam}}^{(i)}}{\hat{\mathbf{d}}_{\text{gaze}}^{(i)}}, \frac{\mathbf{b}_{\text{max}}^{(i)} - \mathbf{o}_{\text{cam}}^{(i)}}{\hat{\mathbf{d}}_{\text{gaze}}^{(i)}} \right)$$
$$\text{Intersection if } t_{\text{near}} < t_{\text{far}} \text{ and } t_{\text{far}} \geq 0, \tag{1}$$

where $\mathbf{o}_{\text{cam}}$ and $\hat{\mathbf{d}}_{\text{gaze}}$ denote the location of the camera and direction of gaze originating from the camera, respectively.

To account for near misses where gaze is directed towards an object but does not intersect its bounding box, we employ a proximity check. First, for a given gaze ray originating at $\mathbf{o}_{\text{cam}}$ with direction $\hat{\mathbf{d}}_{\text{gaze}}$, we find the point on the ray, $\mathbf{p}_{\text{closest}}$, that has the minimum distance to the centre of the object's 3D bounding box, $\mathbf{b}_{\text{centre}}$. This point is found by projecting the vector from the camera to the box centre onto the gaze ray, as shown in Equation 2.

$$t_{\text{closest}} = (\mathbf{b}_{\text{centre}} - \mathbf{o}_{\text{cam}}) \cdot \hat{\mathbf{d}}_{\text{gaze}}$$
$$\mathbf{p}_{\text{closest}} = \mathbf{o}_{\text{cam}} + t_{\text{closest}} \cdot \hat{\mathbf{d}}_{\text{gaze}} \tag{2}$$

From this closest point, we cast a new ray directly towards the bounding box centre, $\hat{\mathbf{d}}_{\text{centre}}$, and use the slab test method to identify where this new ray intersects the box. Specifically, we swap $\mathbf{o}_{\text{cam}}$ for $\mathbf{b}_{\text{centre}}$ and $\hat{\mathbf{d}}_{\text{gaze}}$ for $\hat{\mathbf{d}}_{\text{centre}}$ in Equation 1, yielding a point:

$$\mathbf{p}_{\text{intersect}} = \mathbf{b}_{\text{centre}} + t_{\text{near}} \cdot \hat{\mathbf{d}}_{\text{centre}} \tag{3}$$

A location is considered primed by a near miss if the Euclidean distance, δ, between $\mathbf{p}_{\text{closest}}$ and $\mathbf{p}_{\text{intersect}}$ is below a threshold τ of 5 cm. This threshold was determined empirically: we found that smaller values risked undercounting valid gaze interactions due to minor inaccuracies in gaze or object bounding boxes, while larger values began to accept ambiguous cases. Formally, priming by near miss occurs when:

$$\delta = ||\mathbf{p}_{\text{intersect}} - \mathbf{p}_{\text{closest}}||$$
$$\text{Near miss if } \delta \leq \tau \text{ and } t_{\text{closest}} \geq 0 \tag{4}$$

The second condition in Equation 4 ensures the closest point lies in front of the camera, confirming the user is looking towards the object.

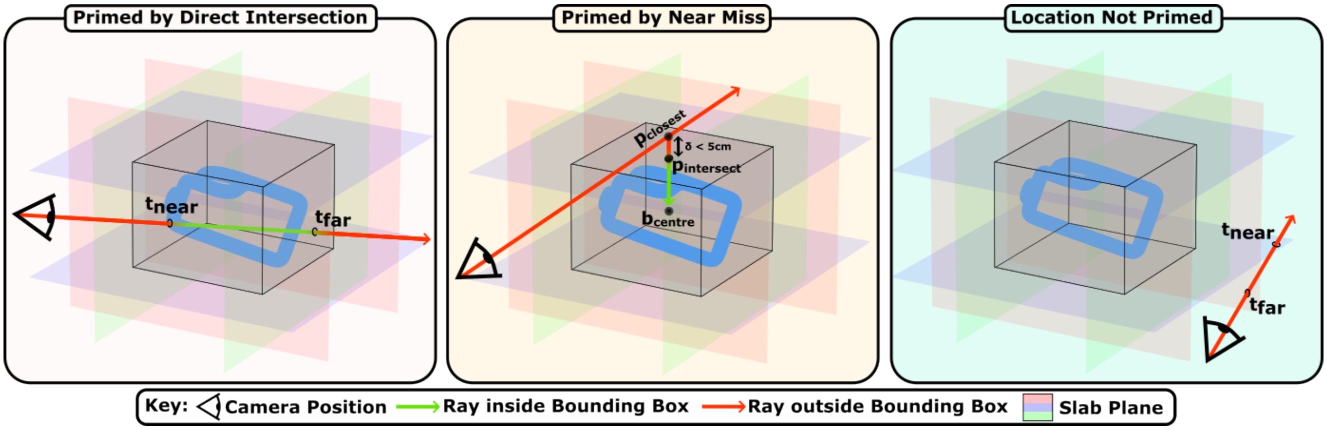


Figure S1. Visualisation of the Slab Test Method [6] for registering primed object interactions. (Left) Primed by Valid Intersection. (Middle) Primed by Near Miss. (Right) No Priming.

We exclude interactions involving only minimal movement ($<$ 20 cm) between the initial pose and goal, as they do not represent meaningful interactions. This filtering process refines the dataset and ensures the quality of P&R sequences so that primed object interactions are not trivial.

B.2. Estimating Full Body Pose for P&R Sequences

Building upon the priming data collected previously, we require full-body pose sequences of primed object interactions. Our generation pipeline uses EgoAllo [11], a method that estimates expressive, full-body human motion from egocentric video and SLAM-based camera poses. The model first converts head pose trajectories into a spatially and temporally invariant representation that encodes relative motion with respect to the ground plane. This representation is used to condition a diffusion-based prior that samples local SMPL-H [9] parameters: pose, representing per-joint rotations over time for the full body including hands; shape, encoding time-invariant body proportions such as height and limb length; and contact predictions, indicating per-joint contact with the environment to improve realism. The model is trained on human motion sequences from AMASS [5], augmented with synthetic egocentric head pose trajectories.

For each interaction, we provide the model with a sequence of video frames and their corresponding camera poses to generate an initial sequence of full-body motions. To enhance the fidelity of hand-object interactions, we calculate the 3D wrist and palm poses from Aria MPS models and provide these to the EgoAllo model to align the generated hands with the wrist and hand locations. This alignment step is crucial; we found that without it, the hands in the generated sequence often remain static and unrealistic. Incorporating these poses yields a more accurate representation of hand orientation in our final motion sequences.

A key design choice in our generation process is the temporal window of the sequences. Specifically, we initiate the generation 2 seconds prior to the moment the object is primed and conclude following the interaction. This decision was made to ensure that our sequences capture any sufficient head motions or other preparatory body movements that precede the explicit eye-gaze priming. By including this anticipatory phase, the resulting sequences provide a more complete and naturalistic depiction of a primed interaction.

B.3. More Statistics of P&R sequences

We show more detailed statistics on each of our curated datasets in Fig. S2. Concretely, the histograms of body movement, hand movement, and prime gap are shown. Body and hand movement measure the maximum displacement of the body or hands within a P&R motion sequence. Prime gap is the duration between the prime time t_p and the pick/put event time t_e.

B.4. Train-Test Splits

For each dataset, we split the source videos into 70% train -30 % test sets. The P&R sequences curated from these videos were automatically distributed to the corresponding subset. HD-EPIC [8] has 156 long videos. We selected 70% (109 videos) for training and the remaining for testing. The curated sequences from the 109 videos were used as train P&R sequences. We perform a similar procedure for MoGaze [4], HOT3D [1], and ADT [7]. Zheng *et al*. [12] proposed a train-test split for GIMO sequences. We use the same split for our curated P&R sequences. Exact train/test split sizes are given in Tab. S1.

C. Ablation of Architecture & Loss

As in the main paper, our ablations are evaluated on HD-EPIC and MoGaze.

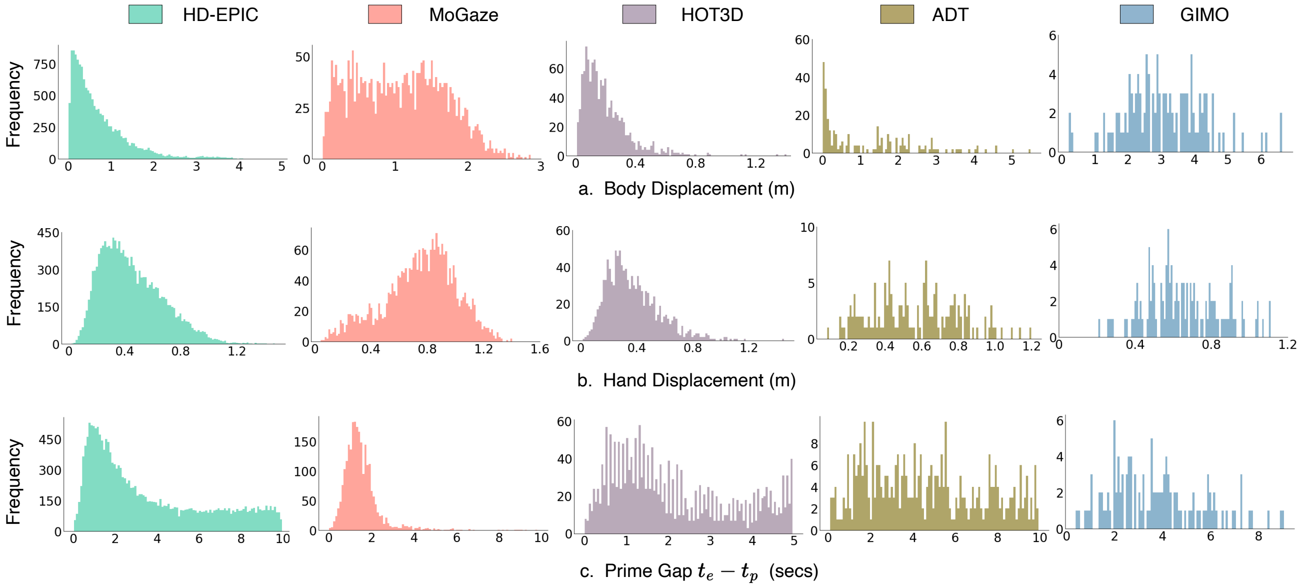


Figure S2. Histograms of body movement, hand movement, and prime gap in curated P&R sequences.

Table S1. **Train/Test splits**. We provide the train-test splits for our curated P&R sequences.

	HD-EPIC	MoGaze	HOT3D	ADT	GIMO
Train	12642	1947	1672	326	108
Test	5492	690	744	85	22
Total	18134	2637	2416	411	130

C.1. Transformer Encoder v/s Decoder

We compare performance of transformer encoder v/s decoder based diffusion model for the task of P&R motion generation in Tab. S2. While the decoder architecture injects condition $\hat{\mathbf{z}}_\mathbf{t}$ by cross-attention with each decoder layer, the encoder provides the condition as an additional token at the input of the first encoder layer. The decoder architecture performs significantly better than the encoder architecture, making it a superior choice for the task.

C.2. Training Loss

We ablate the impact of $\mathcal{L}_{joint}$ in Tab. S3. We find adding the $\mathcal{L}_{joint}$ helps improve P&R generation for both HD-EPIC and MoGaze.

C.3. Incorporating Goal Condition

We pretrain our P&R diffusion model for motion generation only conditioned on text using the Nymeria dataset. For fine-tuning on P&R sequences, we add our initial state and goal pose/target location condition $\mathbf{p}$ to the text condition $\mathbf{z_t}$ to get $\hat{\mathbf{z}}_t$. We ablate another alternative of incorporating $\mathbf{p}$ to $\mathbf{z_t}$ using cross-attention as shown in

$$\delta_\mathbf{t} = CA(\mathbf{z_t}, \mathbf{p}) = \text{Softmax}\left(\frac{(\mathbf{z_t}\mathbf{W}_Q)(\mathbf{p}\mathbf{W}_K)^T)}{\sqrt{d_k}}\right)(\mathbf{p}\mathbf{W}_V)$$

$$\hat{\mathbf{z}}_\mathbf{t} = \mathbf{z_t} + \delta_\mathbf{t} \tag{5}$$

where CA is a 1-layer cross-attention. $\mathbf{z_t}$ is linearly projected to get the query and $\mathbf{p}$ is projected to key and value. We use a residual network to make the most of our pretraining. We show the results in Tab. S4. We find that incorporating the condition through addition performs better, across all metrics.

C.4. Ablation of Total Diffusion Steps

For a single motion generation, the diffusion model starts from noise at $t = T$ and iteratively denoises it over diffusion steps $t = \{T, T-1, \cdots, 0\}$, finally producing the clean motion at $t = 0$. We ablate the choice of T in Tab. S5, which controls the total number of steps needed to generate a sequence of motion. We find that $T = 50$ gives a consistently good performance across all metrics with an improvement of $+1.64\%$ in prime success. Importantly the method is generally robust to the number of steps.

D. Analysing hyper-parameters of Prime Success metric

We conduct an in-depth analysis to better understand the impact of hyperparameters (the time window σ and threshold for angular error θ) used in calculating the newly introduced Prime Success metric. We compare our P&R predictions with the actual gaze while varying the hyperparameters of the metric. Note that as the thresholds are changed,

Table S2. **Encoder v/s Decoder**. We compare encoder and decoder architecture for P&R motion generation.

	HD-EPIC				MoGaze			
Architecture	Prime Success ↑	Reach Success ↑	Loc Err ↓	MPJPE ↓	Prime Success ↑	Reach Success ↑	Loc Err ↓	MPJPE ↓
Encoder	51.59	82.91	22.82	0.29	37.95	56.62	27.90	0.34
Decoder	**59.06**	**89.48**	**16.28**	**0.24**	**41.44**	**93.53**	**22.92**	**0.33**

Table S3. **Loss Ablation**. The improvement by adding the joint loss is evident across both datasets and 7 out of 8 metrics.

	HD-EPIC				MoGaze			
Loss	Prime Success ↑	Reach Success ↑	Loc Err ↓	MPJPE ↓	Prime Success ↑	Reach Success ↑	Loc Err ↓	MPJPE ↓
$\mathcal{L}$	57.70	86.70	16.67	0.31	40.10	86.38	**21.21**	0.34
$\mathcal{L} + \mathcal{L}_{joint}$	**59.06**	**89.48**	**16.28**	**0.24**	**41.44**	**93.53**	22.92	**0.33**

Table S4. **Condition Injection**. We verify different methods for injecting our initial state and goal conditions.

	HD-EPIC				MoGaze			
Method	Prime Success ↑	Reach Success ↑	Loc Err ↓	MPJPE ↓	Prime Success ↑	Reach Success ↑	Loc Err ↓	MPJPE ↓
CA	56.41	85.67	17.33	0.27	39.43	72.62	26.04	0.35
Addition	**59.06**	**89.48**	**16.28**	**0.24**	**41.44**	**93.53**	**22.92**	**0.33**

Table S5. **Impact of diffusion steps** T. We compare the performance of P&R motion generation for multiple diffusion steps.

	HD-EPIC				MoGaze			
T	Prime Success ↑	Reach Success ↑	Loc Err ↓	MPJPE ↓	Prime Success ↑	Reach Success ↑	Loc Err ↓	MPJPE ↓
10	57.42	**90.26**	**16.00**	0.26	41.22	89.88	26.64	0.34
50	**59.06**	89.48	16.28	**0.24**	**41.44**	**93.53**	22.92	**0.33**
100	56.18	85.07	17.38	0.27	38.02	91.29	**22.25**	0.33
500	55.54	85.48	18.13	0.29	36.76	91.96	23.88	0.34
1000	54.68	85.22	17.67	0.29	37.65	90.25	26.04	0.34

the motion can be considered a success or a failure. Recall that our results are reported for $\theta = 16\,\text{deg}$ and $\sigma = 0.2$ sec.

To evaluate the impact of these hyperparameters, we vary θ on the x-axis (between 0 and 90 deg), then plot distinct curves for discrete time windows: 0, 0.2, 0.4, 0.8, and 1.0 seconds. Figure S3 shows that a very tight time window is too restrictive for MoGaze. As expected, a high θ threshold is too permissive and cannot be used to compare different methods.

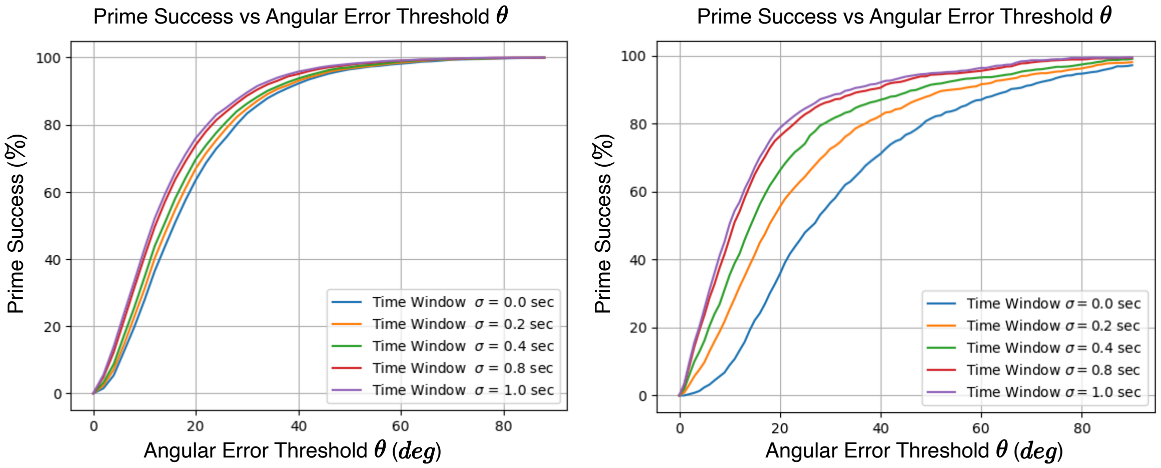
E. Pretraining Results

Here we provide the results of our text-conditioned motion generation pre-training on Nymeria.

As in previous works [10], we train motion-text embedding models [3], with two encoders: one for motion and one for text, using a contrastive loss. We use paired text-motion sequences from the Nymeria train set. We follow the architecture for our encoders from [3].

Following [2, 10], we use the following metrics for evaluation -

- R Precision (Top-3): Given batches of motion and corresponding text, the most similar texts to each motion are ranked based on the Euclidean distances. This calculates the percentage of motion sequences for which the correct text is retrieved in the top 3 matches.
- FID: This compares the encoded feature distribution of the generated motion to that of real motion
- Multimodal Distance: This calculates the average Euclidean distance in the embedding space between paired



a. HD-EPIC b. MoGaze

Figure S3. Varying time window σ and angular error threshold θ for Prime Success calculation on HD-EPIC and MoGaze.

Table S6. **Evaluating encoders**.

Feature Extractor	R Precision (Top- 3) ↑	FID ↓	Multi-modal Distance ↓	Diversity ↑
[10]	25.91 ± 0.17	0	5.16± 0.00	4.26 ± 0.19
Ours	**75.43** ± **0.12**	0	**2.79**± **0.00**	**9.70** ± **0.13**

Table S7. **Pretraining results**.

Pretraining Dataset	Motion	R Precision (Top- 3) ↑	FID ↓	Multi-modal Distance ↓	Diversity ↑
	Real	75.43 ± 0.12	0	2.79± 0.00	9.70 ± 0.13
HumanML3D	Generated	43.32 ± 0.69	11.39 ± 0.64	5.53± 0.08	7.26 ± 0.08
Nymeria	Generated	**77.25** ± **0.42**	**0.97** ± **0.11**	**2.85**± **0.03**	**9.75** ± **0.09**

motion and text.

- Diversity: This measures the variance in the generated motion over all text prompts in the test set.

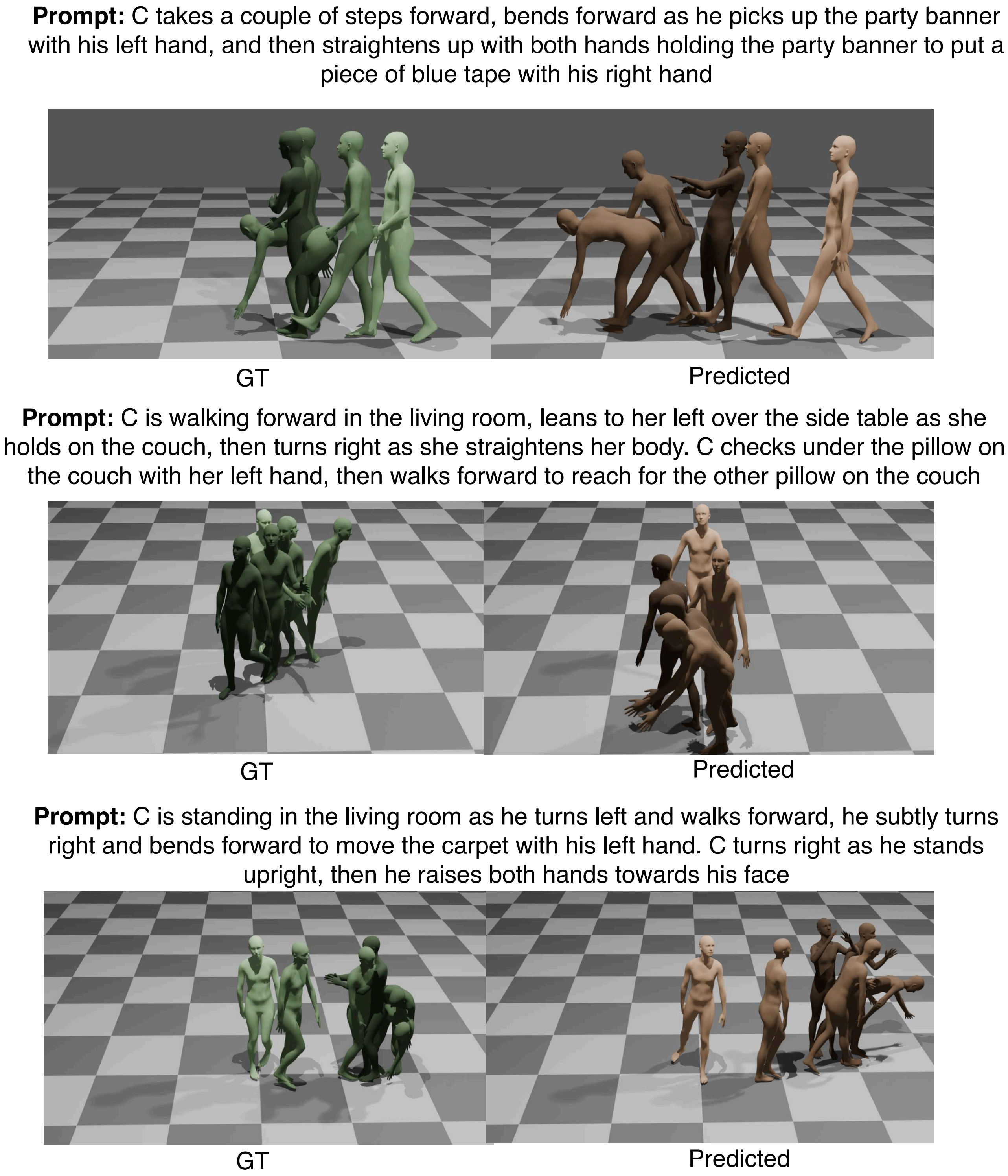
E.1. Evaluating Embedding and Encoders

We first evaluate the quality of our trained encoders. We use the embedding from [10] as our baseline, which is trained on HumaML3D. We compare the encoders on the matched text and real motion pairs of the Nymeria test set in Tab. S6.

Different from HumanML3D, Nymeria text and motion relate to more fine-grained everyday activities *e.g.*, 'In the hallway, C slightly turns her body to the left as she looks at her peer while holding the right side door of the laundry closet using her right hand. Then C slightly turns her body to the right as she looks at the laundry closet and takes her right hand off from the right side door of the laundry closet'. As a result, HumanML3D-trained text embeddings do not work well on the Nymeria test set. The high R-Precision and low multi-modal distance show that our trained embedding on Nymeria better matches text-motion pairs while maintaining high diversity. This shows that our trained embedding is substantially better and we thus use it for evaluating the impact of our pretraining.

E.2. Pretraining Results

We provide the results of our pretrained model in Tab. S7. We find that the motion generated by our Nymeria-pretrained model aligns better with the fine-grained texts of Nymeria. This is verified by the $+34.0\%$ and -2.68 improvements in R-Precision and Multi-modal distance respectively. The diversity of our generated motions is $+2.49$ higher than that of motions generated by the HumanML3D pre-trained model. We showcase some of the qualitative results of the pre-trained model in Fig. S4.



Prompt: C takes a couple of steps forward, bends forward as he picks up the party banner with his left hand, and then straightens up with both hands holding the party banner to put a piece of blue tape with his right hand

Prompt: C is walking forward in the living room, leans to her left over the side table as she holds on the couch, then turns right as she straightens her body. C checks under the pillow on the couch with her left hand, then walks forward to reach for the other pillow on the couch

Prompt: C is standing in the living room as he turns left and walks forward, he subtly turns right and bends forward to move the carpet with his left hand. C turns right as he stands upright, then he raises both hands towards his face

Figure S4. Qualitatives of our pre-trained model. We showcase both ground truth (GT) and predicted motion for given text prompts. Darker poses represent later times.

References

[1] Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Shangchen Han, Fan Zhang, Linguang Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, Jakob Julian Engel, and Tomas Hodan. Hot3d: Hand and object tracking in 3d from egocentric multi-view videos. In *CVPR*, 2025. 2

[2] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *ACMMM*, 2020. 4

[3] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *CVPR*, 2022. 4

[4] Philipp Kratzer, Simon Bihlmaier, Niteesh Balachandra Midlagajni, Rohit Prakash, Marc Toussaint, and Jim Mainprice. Mogaze: A dataset of full-body motions that includes workspace geometry and eye-gaze. *IEEE Robotics and Automation Letters (RA-L)*, 6(2), 2020. 2

[5] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *ICCV*, 2019. 2

[6] Alexander Majercik, Cyril Crassin, Peter Shirley, and Morgan McGuire. A ray-box intersection algorithm and efficient dynamic voxel rendering. *Journal of Computer Graphics Techniques (JCGT)*, 7(3), 2018. 2

[7] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *ICCV*, 2023. 2

[8] Toby Perrett, Ahmad Darkhalil, Saptarshi Sinha, Omar Emara, Sam Pollard, Kranti Parida, Kaiting Liu, Prajwal Gatti, Siddhant Bansal, Kevin Flanagan, Jacob Chalk, Zhifan Zhu, Rhodri Guerrier, Fahd Abdelazim, Bin Zhu, Davide Moltisanti, Michael Wray, Hazel Doughty, and Dima Damen. Hd-epic: A highly-detailed egocentric video dataset. In *CVPR*, 2025. 2

[9] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (TOG)*, 36(6), 2017. 2

[10] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *ICLR*, 2023. 4, 5

[11] Brent Yi, Vickie Ye, Maya Zheng, Yunqi Li, Lea Müller, Georgios Pavlakos, Yi Ma, Jitendra Malik, and Angjoo Kanazawa. Estimating body and hand motion in an ego-sensed world. In *CVPR*, 2025. 2

[12] Yang Zheng, Yanchao Yang, Kaichun Mo, Jiaman Li, Tao Yu, Yebin Liu, C Karen Liu, and Leonidas J Guibas. Gimo: Gaze-informed human motion prediction in context. In *ECCV*, 2022. 2