Supplementary Materials of *Towards Diverse and Natural Scene-aware 3D Human Motion Synthesis*

1. Implementation Details

In this section, we first discuss the detailed structures of the CVAE models used in our framework. Then we discuss the training and inference details of these models.

1.1. Network Architecture.

The encoders and decoders of the action conditioned pose generator in Section 3.2, Place Refiner in Section 3.2, and Neural Mapper in Section 3.3 share the exactly same architectures, which are all two-layer Multilayer Perceptron (MLP). The encoder takes the 256-dim features encoded by the fully-connected layers and predicts the mean $\mu \in \mathbb{R}^{32}$ and the standard deviation $\sigma \in \mathbb{R}^{32}$ for a Gaussian Distribution. We sample the latent code z from this distribution for the decoder during training.

For the motion completion network in Section 3.4, we use the Transformer [12] as the basic structure as [9]. To be specific, we use two fully connected layers to encode all inputs to 256-dimension features. The encoder predicts the mean $\mu \in \mathbb{R}^{32}$ and the variance $\sigma \in \mathbb{R}^{32}$ for a Gaussian Distribution as the CVAE model for action conditioned poses. Following [9], we set 8 layers of the Transformer network for the encoder and decoder.

1.2. Training and Inference Details.

Scene-Agnostic Pose Synthesis. Firstly, we show how to train the CVAE model for scene-agnostic pose synthesis in Section 3.2. As the standard VAE [7] model, the training objective consists of two parts. The first one is the reconstruction loss between the reconstructed human poses and the input human pose. The other objective is Kullback-Leibler (KL) Divergence between the Gaussian Distribution $Q(z|\mu, \sigma)$, where μ and σ are predicted by the encoder, and the standard Gaussian Distribution $N(0, I^2)$.

Place Refiner. The Place Refiner takes the placed body poses, and the scene contexts encoded by the PointNet [10] as inputs and predict the offset $\Delta t_i, \Delta o_i$ for the sampled $\bar{t}_i, \bar{o}_i$. To train this network, we first build up the discrete candidates following the same procedure of scene-conditioned anchor placing in Section 3.2 for each scene in our training set. For practice, we split each scene into non-overlapping discrete grids uniformly as translation candidates and then uniformly sample eight different orientations paralleling with the ground plane as the orientation candidate. Each pose in the given scene is neighbor to four-position candidates. We assign each pose in the training set to one randomly sampled neighbor position candidate and one orientation candidate of this position candidate. Then we predict the offset from these candidates to the original translation and orientation for this pose. Similar as [7], the training objective is the reconstruction loss and the KL-Divergence.

Neural Mapper. The input of Neural Mapper includes the local context encoded by BPS [15] and the human moving direction in this local context, which is obtained from ground-truth moving paths of PROX [4] dataset. To train this model, we first split the motions in the training set of the PROX dataset into different 60 frame sequences. The local context is cropped as a $2m \times 2m \times 2m$ cubic cage at the motion center as [15]. To compute BPS features, we uniformly sample a set of $N_b = 10^4$ basis points within the unit sphere at the center of the local context and then normalize the local scene context into the same unit sphere. The final BPS feature is the concatenation of these minimal distances $\mathbf{x}_s \in \mathbb{R}^{N_b \times 1}$ between the sampled unit sphere and normalized scene context. In practice, we use two additional fully-connected layers to further encode $\mathbf{x}_s$ as the local context feature. Similar to the standard VAE [7], the training objective of Neural Mapper consists of a KL-Divergence term and a reconstruction term. Specially, we norm the moving direction between the beginning and ending points of the motion sequence to $[0, 1]$ as the reconstruction target during training. The reconstruction term estimates the residuals between this normalized moving direction and the expectation of the estimated direction distribution.

Path Refiner and Motion Synthesizer. We train our Path Refiner and Motion Synthesizer together in an end-to-end manner. Both two models synthesize $M = 60$ frames of paths or motions. Similarly, the training objective consists

of the reconstruction loss on the synthesized paths or motions, as well as the KL-Divergence. Specially, we do not use the planned path obtained from Section 3.3 in training the Path Refiner. Instead, we use the directions pointing from the beginning to the ending point of the motion as the planned path for practice. The reason mainly lies in two aspects. The first one is that repeat running of path planning module to obtain planned paths is not efficient in training. The second one is that the shortest path from the planning module is similar to the straight line in short-term motions.

During the inference stage, we find that directly synthesizing motions from two consecutive anchors lead to unstable results. We believe it is majorly caused by the variance of the lengths of the planned paths. To resolve this, we first split the planned paths into several pieces with equal length. Each split point is then assigned with an intermediate status action label. Using the new action labels, the intermediate anchors can be produced following the same method as placing human-scene interaction anchors in Section 3.2. Given the new anchors and the split paths, our motion completion network synthesizes human motions for each piece and then connects them together as the integrated motion. For practice, we insert the motions with random poses conditioned on "walking", "standing", and " squatting" action.

1.3. Optimization.

We perform optimizations to improve the motion quality with the motion and physics constraints. For example, the human should walk on the floor with smooth motions. We conduct the optimization in [5] and [13] for human-scene interaction anchors and motion sequences, respectively. For better human-scene interaction anchors, we use the objective functions defined in [5], that consist of the affordance loss for contacting the specific body parts to the given scene (*e.g.* foot to the floor), penetration loss for the reasonable physical relationship between body meshes and the reconstructed SDF (sign distance field), and the regularization to keep the optimized pose close to the initial pose. We optimize all these human-scene interaction anchors for 10 iterations with $1e{-}3$ learning rate, using L-BFGS [8] algorithm as [5]. For the synthesized motion obtained in Section 3.4, we follow their optimization objective functions [13] for foot location, environment, and motion smoothness to improve the motion quality. We optimize all our motions for 100 iterations with $1e{-}2$ learning rate, using ADAM [6] algorithm as [13].

2. Experiments

Naturalness Results on PROX. Then we compare the naturalness of these methods in Table 3. For the physical plausibility, we use the same motion as the comparison in Table.3 of our paper. For user study, we randomly sample motions with 2, 4, and 8 different target actions. All the comparison results show that our method can synthesize more natural motions than other methods do. Especially, our method achieves better results without the optimization post-process, because of the guidance from planned obstacle-free paths. The comparison between the last two rows shows that the guidance of the proposed Path Refiner is advantageous in synthesizing natural motions.

Table 1. **Evaluation on naturalness of synthesized motions on PROX [4].** We measured this by physical plausibity (non-collision and contact score) as well as user study.Specially, w/ and w/o opt means the results with/without optimization post-process [13]. "Ours*" means our motion completion network without the Path Refiner.

Method	Non-Collision ↑		Contact ↑		User Study ↑
	w/o opt	w/ opt	w/o opt	w/ opt	
SA-CSGN [14]	92.37	98.21	95.36	98.72	2.74(0.97)
Wang et.al. [13]	93.88	98.72	96.42	99.35	3.42(1.06)
SAMP [3]	94.92	99.31	96.28	99.32	3.46(0.96)
Ours*	94.52	99.28	96.24	99.27	3.28(0.94)
Ours	**95.93**	**99.61**	**96.45**	**99.35**	**3.68(0.84)**

Table 2. **Evaluation on human-scene interaction anchors for Matterport3D [2].** We evaluate the diversity of the human-scene interaction anchors (Anchor, considering θ, t, and ϕ) and the placing (Position, considering only t and ϕ) with/without optimization post-process. S means the sampling strategy based on pose relationship in Section 3.2, and R means our Placing Refiner.

Method	Anchor		Position	
	Entropy ↑	Cluster ↑	Entropy ↑	Cluster ↑
Baseline [5]	2.54 / 2.50	2.45 / 2.44	2.51 / 2.50	0.58 / 0.56
Baseline [5] + S	2.63 / 2.61	2.53 / 2.53	2.54 / 2.54	0.64 / 0.65
Baseline [5] + S + R	**2.70 / 2.68**	**2.59 / 2.58**	**2.68 / 2.66**	**0.72 / 0.72**

Table 3. **Evaluation on synthesized motion for Matterport3D [2].** Comparison on **APD**, **non-collision** score and **contact** score on Matterport3D dataset. Specially, "w/ OPT" and "w/o OPT" refer to the results obtained with/without optimization post-process [13]. "Ours*" means our motion completion network without the Path Refiner.

Method	APD ↑		Non-Collision ↑		Contact ↑	
	w/o OPT	w/ OPT	w/o OPT	w/ OPT	w/o OPT	w/ OPT
SA-CSGN [14]	2.24	2.26	91.51	99.08	95.21	99.33
Wang et.al. [13]	0.00	0.00	93.78	99.42	96.48	99.35
SAMP [3]	2.46	2.48	94.35	99.32	96.46	99.35
Ours*	2.34	2.38	94.12	99.08	96.46	99.32
Ours	**2.57**	**2.60**	**95.72**	**99.42**	**96.72**	**99.36**

Results on Matterport3D. We show the quantitative results on Matterport3D dataset [2]. The sampling strategies are the same as our experiments on PROX dataset. We first perform **K-Means** ($K = 20$) and evaluate the obtained human-scene interaction anchors on Matterport3D with the entropy of cluster sizes and the average distance between the cluster center and the samples belong to it.

As shown in Table 2, our method enhance the diversity of the anchors for motion synthesis. Besides, we evaluate

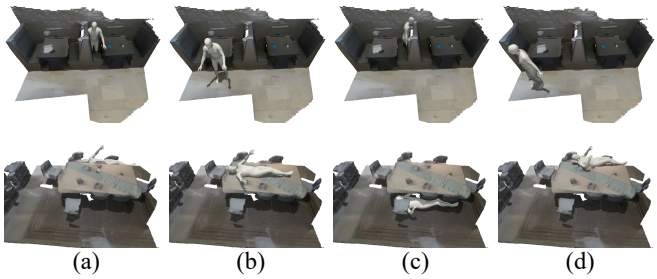

Figure 1. **Failure cases of the scene-centric paradigm.** We use the action label as additional condition to extend previous scene-centric paradigms [15, 16]. The first and third column shows results obtained from [16] and [15], respectively. The second and forth columns shows the results generated by our framework with the same pose as the first and third columns, respectively.

the synthesized motion on Matterport3D via the **APD**, **Non-Collision** score and **Contact** score. The results are listed in Table 3. It is revealed that our method can synthesize better results than previous methods with better diversity and physical plausibility. Besides, the Path Refiner still can improve the diversity and naturalness on this dataset.

3. Further Discussion

In this section, we first discuss the reason for using the human-centric paradigm, for human-scene interaction anchors. The human-centric paradigm means we place the sampled poses to the positions which match the physical structure of these poses. Then we show how to use our Neural Mapper to work with other manually set constraints. At last, we show the influence of the planned path on motion synthesis.

Human-Scene Interaction Anchor. Previous works [15, 16] of synthesizing human-scene interaction anchors aim to explore the influence of scene context to place human pose in the given scene and neglect the action labels. Intuitively, we can incorporate these action labels as an additional condition and incorporate them into their frameworks to synthesize poses. However, as shown in Figure 1, simply extending the previous works cannot guarantee to synthesize the physically plausible poses with the given actions and scene contexts. We believe it is due to the reason that these methods do not build up the relationship between the action and the scene context explicitly. For example, method [16] directly uses the pooled 2D image features as the condition and ignores the relationship between the spatial information and the action. Another method [15] first samples different positions to build up the BPS and then synthesize different poses. However, the poses for each action have their specific physical structure and match different scene structures. It is difficult to find suitable places for the poses conditioned on the given action label, as shown in the Figure 1.

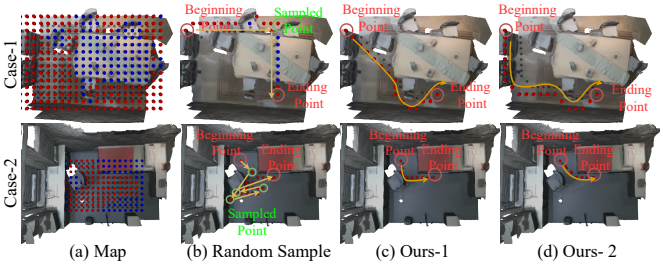

Figure 2. **Comparison with randomly sampled intermediate points.** Our method can plan diverse and natural paths without complex manual constraints.

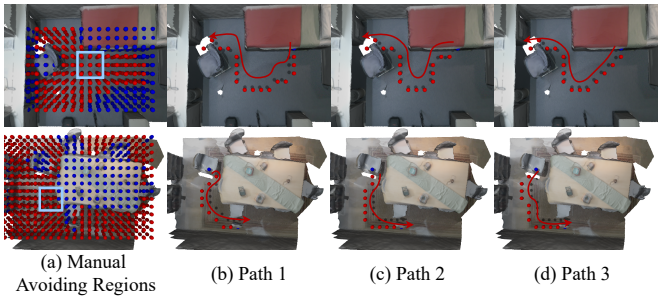

Figure 3. **Neural Map with manual constraints.** We change the valid grids in the blue box as the manual avoiding regions. These regions are the shortest path from these two points. We find that our Neural Map can work with this constraint to sample different planned paths.

Instead, the human-centric paradigm proposed by us can effectively leverage the explicit relationship between the synthesized 3D human and the scene structure (*e.g.* physical and semantic structures) and thus makes the whole placing process more controllable.

Neural Mapper Several failure cases generated from randomly sampling intermediate points are included in Figure 2. In the first row, when sampled intermediate points and the ending points are obstructed, the original $A^\star$ algorithm can not find paths for these points. We adjust $A^\star$ by allowing to search paths in the obstacle regions , and $A^\star$ only produces impractical paths crossing the table as the first row of Figure 2. In the second row, random sampled points can also lead to unnatural zigzag paths. One may argue that these failures can be avoided via complex constraints used in previous methods [1, 11]. However, the proposed Neural Mapper provides an **automatic and data-driven** way to embed semantic information into natural and diverse path planning, without complex constraints. Besides, our Neural Mapper also can work with manually set constraints, such as avoiding passing a certain region. We show the planned results in Figure 3. It is revealed that our method can still produce natural and diverse paths under such constraints.

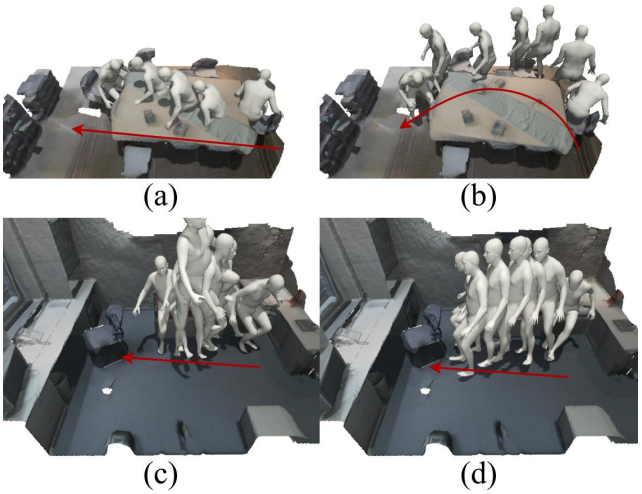

Figure 4. **Effect of the planned path.** (a) is the result without planning module and (b) is based on planning module. (c) is the results for only using translations of planned path as the position encoding for our Path Refiner in Section 3.4 and (d) is result from our method.

Planned Path for Motion Synthesis. As shown in Figure 4, we show the effectiveness of using planned paths in the procedure of motion synthesis. Firstly, without the additional positional encoding from the planned path, the synthesized motion can not follow the planned path and penetrate to the table, as shown in Figure 4 (a). Besides, we find that both the translation and orientation for the planned path are also crucial for motion synthesis. As shown in Figure 4 (c), the Path Refiner synthesizes unnatural orientations for human motions without encoding the orientation of the planned path into the positional encoding as Section 3.4. Instead, as shown in Figure 4 (b) and (d), our method can synthesize natural human motion with the translation and orientation of planned path as the additional positional encoding for our Path Refiner.

References

[1] Stefano Carpin. Randomized motion planning: a tutorial. *International Journal of Robotics and Automation*, 2006. 3

[2] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *International Conference on 3D Vision (3DV)*, 2017. 2

[3] Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael Black. Stochastic scene-aware motion prediction. In *Proceedings of the International Conference on Computer Vision 2021*, Oct. 2021. 2

[4] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black. Resolving 3D human pose ambiguities with 3D scene constraints. In *International Conference on Computer Vision*, 2019. 1, 2

[5] Mohamed Hassan, Partha Ghosh, Joachim Tesch, Dimitrios Tzionas, and Michael J. Black. Populating 3D scenes by learning human-scene interaction. In *Conference Computer Vision and Pattern Recognition*, 2021. 2

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2014. 2

[7] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1

[8] Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 1989. 2

[9] Mathis Petrovich, Michael J. Black, and Gül Varol. Action-conditioned 3D human motion synthesis with transformer VAE. In *International Conference on Computer Vision (ICCV)*, 2021. 1

[10] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. 1

[11] Petr Švestka and Markus Hendrik Overmars. Probabilistic path planning. 1998. 3

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, 2017. 1

[13] Jiashun Wang, Huazhe Xu, Jingwei Xu, Sifei Liu, and Xiaolong Wang. Synthesizing long-term 3d human motion and interaction in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[14] Jingbo Wang, Sijie Yan, Bo Dai, and Dahua Lin. Scene-aware generative network for human motion synthesis. In *Conference on Computer Vision and Pattern Recognition*, 2021. 2

[15] Siwei Zhang, Yan Zhang, Qianli Ma, Michael J. Black, and Siyu Tang. PLACE: Proximity learning of articulation and contact in 3D environments. In *International Conference on 3D Vision (3DV)*, Nov. 2020. 1, 3

[16] Yan Zhang, Mohamed Hassan, Heiko Neumann, Michael J. Black, and Siyu Tang. Generating 3d people in scenes without people. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 3