Stochastic Scene-Aware Motion Prediction
Supplementary Material

Mohamed Hassan[1] Duygu Ceylan[2] Ruben Villegas[2] Jun Saito[2] Jimei Yang[2] Yi Zhou[2] Michael Black[1]
[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany [2]Adobe Research
{mhassan, black}@tue.mpg.de {ceylan, villegas, jsaito, jimyang, yizho}@adobe.com

Figure 1: Examples of action styles in our motion capture data.

Labels	Minutes	Percentage %
Idle	18.3	17.7
Walk	42.3	41.0
Run	5.1	4.9
Sit	27.3	26.4
Lie down	10.1	9.7
Total	103.3	

Table 1: Motion capture data breakdown with respect to actions.

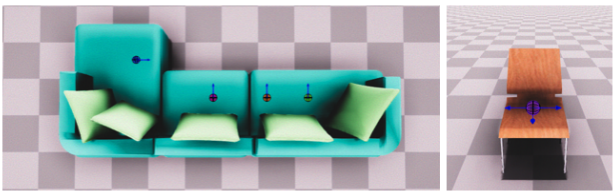

Figure 3: Goal Labelling.

Category	Number of Objects
Armchairs	15
Chairs	16
Sofa	20
L-Sofa	18
Tables	18
Total	87

Table 2: GoalNet data breakdown with respect to object categories.

1. Data Preparation

1.1. Motion Data

Fig. 1 shows examples of different sitting and lying down styles from our MoCap. A breakdown of the dataset in terms of different actions is shown in Table 1. The objects used during the MoCap are shown in Fig. 2.

Figure 2: Objects used during motion capture.

1.2. Goal Data

We select 5 categories from ShapeNet namely, sofas, L-shaped sofas, chairs, armchairs, and tables. From each category, we select $15-20$ instances and we manually label $1-5$ goals for each instance. Table. 2 shows the number of instances for each category. We manually label $1-5$ goals for each instance. The number of goals labelled per instance depends on how many different goals an object can afford. For example, we label 5 different goals for the L-shaped sofa compared to 3 for the chair as shown in Fig. 3.

Network	Architecture
State Encoder	$\{512, 256, 256\}$
Interaction Encoder	$\{256, 256, 256\}$
Gating Network	$\{512, 256, 12\}$
Prediction Network	$\{512, 512, 647\}$

Table 3: Architecture details. All networks are all three-layer fully connected networks with ELU.

2. Training Details

2.1. MotionNet

The character state $\boldsymbol{X}$ is of size 647. The State Encoder, Interaction Encoder, Gating Network, and Prediction Network are all three-layer fully connected networks with rectified linear function ELU. The dimensions of each network are in Table 3. The encoder latent code $\boldsymbol{Z}$ is of size 64 and we set the number of experts K to 12. We use a learning rate of $5e-5$ and train our network for 100 epochs. We use the Adam optimizer with linear weight decay. The weight of the Kullback-Leibler divergence β_1 is 0.1.

2.2. GoalNet

The Interaction Encoder of GoalNet is a three-layer fully connected network of shape $\{512, 512, 64\}$. The latent vector $\boldsymbol{Z}_{goal}$ is of size 3. The weight of the Kullback-Leibler divergence β_2 is 0.5. We use the Adam optimizer with a learning rate of $1e-3$ and train GoalNet for 100 epochs.

2.3. Schedule Sampling

For the schedule sampling training strategy, we set $C_1 = 30$ and $C_2 = 60$. We define a roll-out window of size L where we set $L = 60$ in our experiments. For each roll-out, we feed the ground truth first frame as input to the network and then sequentially predict the subsequent frames while using the scheduled sampling strategy. We divide our training data to equal-length clips of size L.

3. Baselines

As our baselines, we choose a feedforward network (MLP) and a Mixture of Experts (MoE). The architecture of the MLP is shown in Fig. 4. We use the same Interaction Encoder used for our MotionNet followed by four fully connected layers of size 512. The architecture of the MoE is shown in Fig. 5. The Interaction Encoder, Gating Network, and Prediction Network are all the same as the one used in MotionNet.

4. Schedule Sampling

We found that using Schedule Sampling is essential to enable the character to successfully reach the goal and execute the action. Without it, we found the model to often diverge, get stuck, or take very long time to reach the goal as we show in Fig. 6.

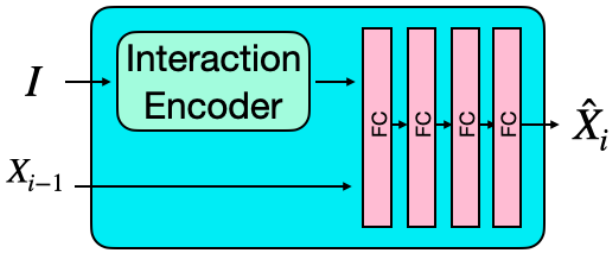

Figure 4: MLP Architecture.

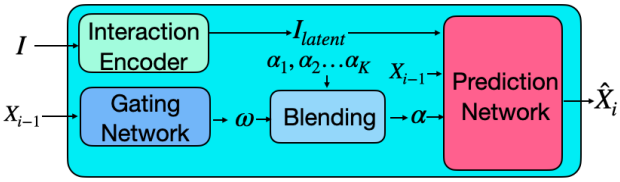

Figure 5: MoE Architecture.

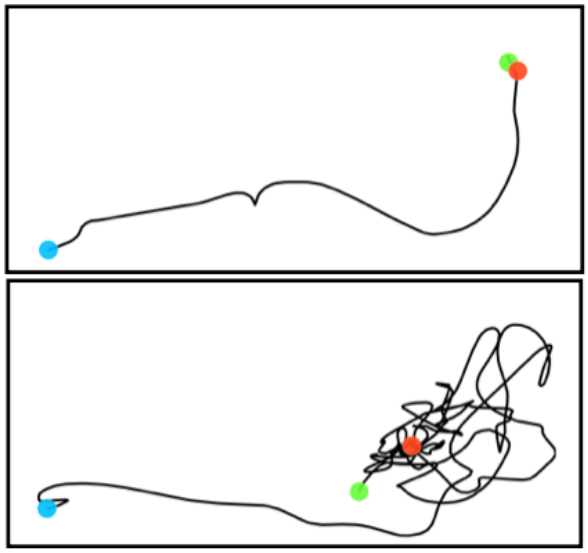

Figure 6: SAMP With Schedule Sampling (Top) and without (bottom). The black line shows the root projection on the xz plane. The blue and green circles denote the root at the first and last frame respectively. The red circle denotes the goal position. Note how SAMP fails to reach the goal without the use of Schedule Sampling.

5. Path Planning Formulation

In order to use the Path Planning Module, we first compute the surface area where the character could stand or

move. We call this the *navigation mesh*. This is computed from the character cylinder collider and the scene geometry. The *navigation mesh* is stored as convex polygons. To find a path between given start and end points, we first map these points to the closest polygons and then use A* to find the shortest path between the polygons [1].

6. Data Augmentation Details

When the object is transformed, the contacts follow the same transformation. When the object is replaced by a new one, we project the original contact by finding the closest points on the surface of the new object. The new motion curve is computed by interpolation and the whole full body pose is computed using CCD IK solver. This does not guarantee smoothness but we found it to be stable in practice. More details are in [2].

7. Interaction Encoder Ablation:

To quantify the importance of the Interaction Encoder, we trained SAMP without the Interaction Encoder. We found that the precision of reaching the goal deteriorates to 14.82 cm and 3.65 deg compared to 6.09 cm and 3.55 deg when the Interaction Encoder was used.

8. Comparison to Cao et al.:

While relevant, the formulation of Cao [1] et al. is significantly different than our method making a direct comparison difficult. Given a target interaction object and action (e.g. "sit on the couch"), SAMP samples a goal location and orientation on the object, computes an obstacle-free path towards the object, and synthesizes diverse motion sequences that are of arbitrary length until the goal is executed. We assume that the character starts the action from an idle position without any knowledge of the past. In contrast, Cao et al. sample a goal *location* in the image space given a one-second-long *history* of motion. Based on this trajectory, a deterministic motion sequence of fixed length (two-seconds) is synthesized. The action executed in this trajectory is not controllable.

9. Failure Cases

We observe that SAMP might not adapt well to objects with significantly different geometry than those seen in training as shown in Fig. 7. Future work might explore different methods of encoding the object geometry.

Figure 7: SAMP with significantly different geometry.

References

[1] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qi-Zhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In *European Conference on Computer Vision (ECCV)*, pages 387–404. Springer, 2020. 3

[2] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Trans. Graph.*, 38(6), Nov. 2019. 3

[1] https://docs.unity3d.com/Manual/nav-InnerWorkings.html