Long-term Human Motion Prediction with Scene Context

Zhe Cao[1,3], Hang Gao[1], Karttikeya Mangalam[1], Qi-Zhi Cai[3], Minh Vo[2], and Jitendra Malik[1]

[1] UC Berkeley
[2] Nanjing University
[3] Facebook Reality Lab

In the supplement, we first compare relevant datasets in Section 1. We also cover the implementation details in Section 2. Further, we evaluate GoalNet in Section 3 and long-term predictions on PROX in Section 4. We also show qualitative comparison results in Section 5. Finally, we describe the network architecture in details in Section 6. We release our dataset and include more visual results in https://people.eecs.berkeley.edu/~zhecao/hmp/.

1 Dataset Comparison

Dataset	#Clips	#Frames	#View per clip	#Characters	#Scenes	Pose jittering	Depth Range	3D scene
H3.6M [2]	80	3.6 M	4	11 (6M 5F)	1		0 – 6	
PiGraph [8]	63	0.1 M	1	5 (4M 1F)	30	✓	0 – 4	✓
PROX [1]	60	0.1 M	1	20 (16M 4F)	12	✓	0 – 4	✓
Ours	119	1 M	14-67	50 (25M 25F)	49		0 – 200	✓

Table 1: Overview of the publicly available datasets on 3D human motion.

In Table 1, we list some representative datasets on 3D human motion including Human3.6M (H3.6M) [2], PiGraphs [8], and Proximal Relationships with Object eXclusion (PROX) [1]. H3.6M [2] is a large-scale dataset with accurate 3D pose annotations using a multi-camera capturing system. However, all recordings were captured in the lab environment (mostly empty space) and thus it lacks diverse human interaction with the indoor environment, e.g., sitting on a sofa or climbing stairs. PiGraphs [8] and PROX [1], on the other hand, are dedicated datasets with extensive efforts for modeling the interaction between 3D human and 3D scene. Due to extensive efforts required for manually collecting the RGBD sequence of human activities, both datasets have a relatively small number of frames, scenes, and characters. They are also less diverse in terms of camera poses and background appearance (only one static camera viewpoint for each entire video clip). As shown in our experiments (Table 1 in the main paper), models trained on these datasets tend to be overfitting to the training data. Their 3D human poses are also relatively noisy, e.g., temporal jittering, due to the difficulty of obtaining accurate 3D poses in the real-world setting.

In contrast, we collect a large-scale and more diverse dataset with clean annotations by developing an automatic data collection pipeline based on the gaming engine. We diversify the camera location and viewpoint over a sphere around the actor such that it points towards the actor. We use in-game ray tracing API and synchronized human segmentation map to track actors. When the actor walks outside the field of view, the camera will be resampled immediately. We believe our synthetic dataset with clean annotations can be complementary to real data for stable training and rigorous evaluation.

2 Implementation Details

In this section, we describe the implementation details for each module of the model. All modules are implemented in PyTorch [6] and trained using the ADAM optimizer [3]. We set the input image size and the heatmap size to 256×448; the resolution of output future heatmap prediction to 64×112; all depth dimension values are caped by 10 and normalized by a factor 4 during training. We train all 3 modules separately and find it works better than joint training. The multi-modal nature of this problem makes it hard to train the model with intermediate prediction, e.g., it is not quite reasonable to supervise PathNet with ground-truth (GT) 3D path towards the GT destination, when taking the input of a very different destination predicted by GoalNet.

GoalNet: We use a 10^{-4} learning rate without weight decay. For both datasets, we train for 2 epochs with a batch size of 128.

PathNet: We train our PathNet with ground-truth destination input, while during inference, we use the prediction from GoalNet instead. The learning rate is set to 2.5×10^{-4} with a 10^{-4} weight decay. Our models are trained for 10 epochs for GTA-IM and 6 epochs for PROX where learning rates decay by a factor of 10 at 7th and 4th epochs, respectively. We use a batch size of 32.

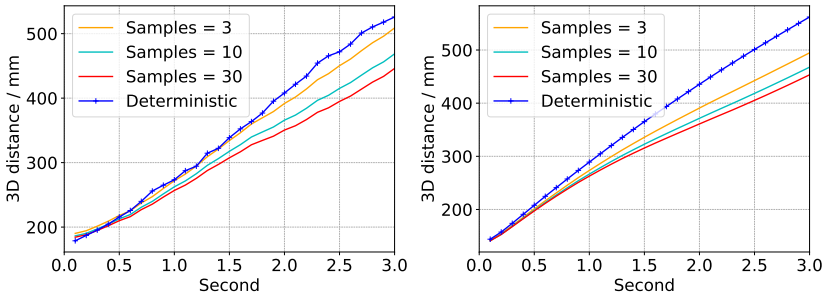
PoseNet: We train the PoseNet ground-truth 3D path, while during inference, we use the prediction from PoseNet instead. We train the model for 80 epochs using a learning rate of 1×10^{-3}, an attention dropout rate of 0.2, and batch size 1024.

3 GoalNet Evaluation

In Table 2, we evaluate 2D future destination predictions of GoalNet. We use the metric of Mean Per Joint Position Error (MPJPE) [2] in the 2D image space. We compare the stochastic results sampled from GoalNet with the deterministic results. We vary the number of samples during the evaluation and present results on both datasets. Our findings are twofold. (1) Directly predicting 2D destinations is beneficial. Our GoalNet can achieve similar performance with the deterministic baseline using as few as 5 samples on both datasets. (2) With more samples, our prediction performance increases monotonously. When using

	GTA-IM dataset		PROX dataset	
Dataset	min	avg$_{\pm std}$	min	avg$_{\pm std}$
Ours (deterministic)	23.7	-	27.7	-
Ours (samples=3)	25.3	41.6$_{\pm 10.3}$	30.3	38.3$_{\pm 7.1}$
Ours (samples=5)	23.6	40.3$_{\pm 12.7}$	27.7	37.2$_{\pm 8.3}$
Ours (samples=10)	17.6	34.6$_{\pm 14.0}$	24.9	35.3$_{\pm 9.1}$
Ours (samples=30)	**12.2**	35.4$_{\pm 17.2}$	**21.7**	31.3$_{\pm 9.7}$

Table 2: **Evaluation of 2D goal prediction results in both dataset**. We compare our results of directly predicting 2D destination using GoalNet with those obtained by our deterministic PathNet. We compare them in terms of the least and average error among all samples.



(a) predicted 3D paths (b) predicted 3D poses

Fig. 1: **Comparison between our stochastic predictions and deterministic predictions of long-term prediction on PROX.** We show error curves of predicted (a) 3D paths and (b) 3D poses with varying numbers of samples over varying timesteps on GTA-IM dataset. In all plots, we find that our stochastic model can achieve better results with a small number of samples, especially in the long-term prediction (within 2-3 seconds time span).

30 samples, our GoalNet can outperform the deterministic baseline by large margins, bringing around 40% less error on GTA-IM dataset and 20% less error on PROX dataset.

4 Long-term Evaluation on PROX

In Figure 1, we evaluate long-term prediction results on PROX dataset as we previously showed on GTA-IM dataset. Specifically, we show results on predicted 3D paths and predicted 3D poses using our deterministic model and stochastic model with varying numbers of samples. We note similar trends as previously seen on GTA-IM dataset. More interestingly, we find that our stochastic mod-

els can beat their deterministic counterpart using only 3 samples on PROX, compared to 5 samples on GTA-IM.

5 Qualitative Results

We show additional qualitative comparison results in Figure 2 and our long-term stochastic predictions in Figure 3. More results on 3D path and 3D pose prediction are in our **video** which can be found at https://people.eecs.berkeley.edu/~zhecao/hmp/.

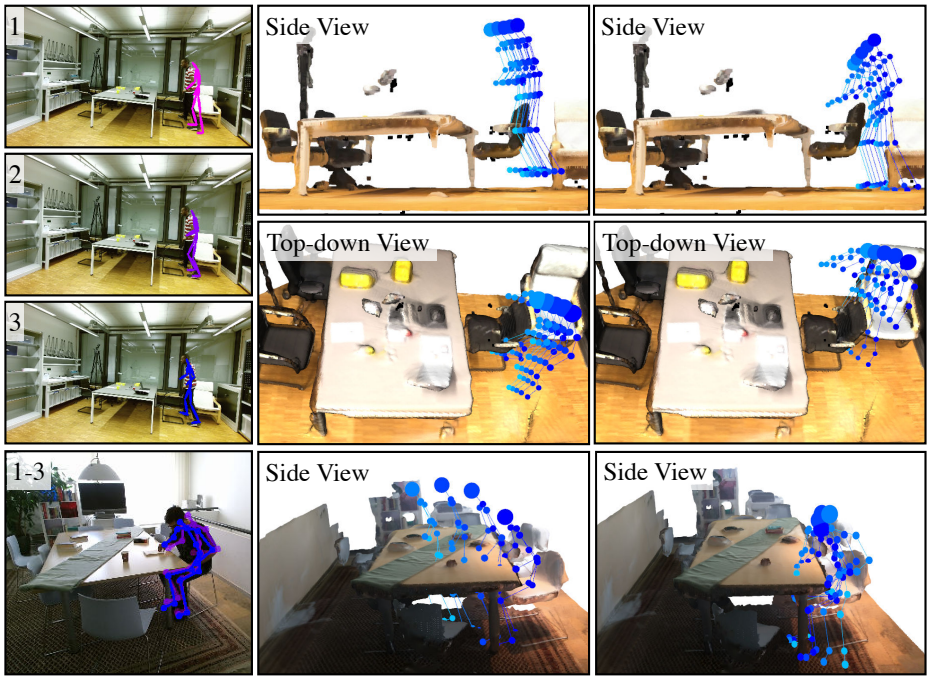


Fig. 2: **Qualitative comparison**. We visualize the input (a), the results of VP[7] and LTD [10] (b) and our results (c) in the ground-truth 3D mesh. The color of pose is changed over timesteps from purple to dark blue and finally light blue. The first example (the 1st and 2nd row) includes both top-down view and side view of the results. From the visualization, we can observe some collisions between the baseline results and the 3D scene, while our predicted motion are more plausible by taking the scene context into consideration.

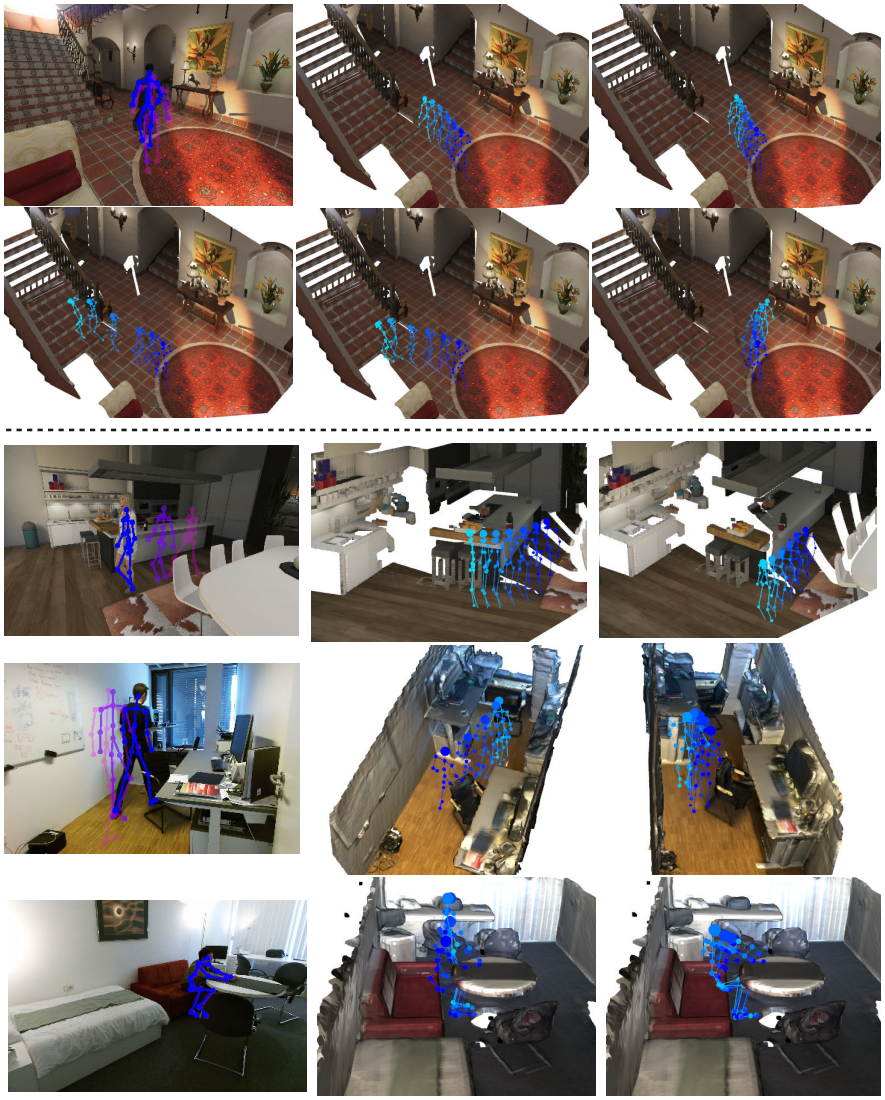


Fig. 3: **Qualitative results on long-term stochastic prediction**. In each example, we first show the input image with 2D pose histories and then our stochastic predictions. In the first example (1st and 2nd row), we show five different future human movement predictions by sampling different human "goals". Depending on his intention, the person can choose to turn left to climb upstairs; he may also go straight through the hallway or turn right to fetch some items off the table. For the following each row, we show two stochastic predictions per example. Our method can generate diverse human motion, e.g., turning left/right, walking straight, taking a u-turn, standing up from sitting, and laying back on the sofa.

6 Network Architecture

We outline our network architectures in this section. Specifically, we define our PathNet in Table 3, GoalNet in Table 5. For PoseNet, please refer to [9], we modified the architecture by removing the input embedding layer, output embedding layer, positional encoding layer and the softmax layer.

Index	Input	Data	Operator	Output shape
(1)	-	scene image	-	$3 \times 256 \times 448$
(2)	-	stacked heatmaps	-	$(N \times J) \times 256 \times 448$
(3)	-	goal heatmap	-	$1 \times 256 \times 448$
(4)	-	initial depth.	-	$N \times 1 \times 1$
(5)	-	2D pose sequence	-	$N \times J \times 2$
(6)	(1), (2), (3)		7×7, stride 2	$128 \times 128 \times 224$
(7)	(6)		$\begin{bmatrix} 3 \times 3, \text{stride } 2 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$256 \times 64 \times 112$
(8)	(7)	backbone feat_1.	HourglassStack	$256 \times 64 \times 112$
(9)	(8)	backbone feat_2.	HourglassStack	$256 \times 64 \times 112$
(10)	(9)	backbone feat_3.	HourglassStack	$256 \times 64 \times 112$
(11)	(8) or (9) or (10)		$\begin{bmatrix} 3 \times 3, \text{stride } 1 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$256 \times 64 \times 112$
(12)	(11)	heatmap pred.	1×1, stride 1	$T \times 64 \times 112$
(13)	(8) or (9) or (10)		$\begin{bmatrix} 3 \times 3, \text{stride } 2 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$384 \times 32 \times 56$
(14)	(13)		$\begin{bmatrix} 3 \times 3, \text{stride } 2 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$512 \times 16 \times 28$
(15)	(14)		GlobalAvgPool	$512 \times 1 \times 1$
(16)	(4), (5)		Linear	$256 \times 1 \times 1$
(17)	(15), (16)		Linear	$256 \times 1 \times 1$
(18)	(17)	depth pred.	Linear	$(N + T) \times 1 \times 1$

Table 3: **Overall architecture for our PathNet.** Each convolutional block denoted in the bracket has an internal skip connection with appropriate strides. Each convolutional operator is followed by a batch normalization and ReLU layer, except the one before heatmap prediction. Each linear operator is followed by a layer normalization and ReLU layer, except the one before depth prediction. We denote N as input time frames, T as output time frames, J as the number of human keypoints. We obtain initial depth as input by scaling the size of the human bounding box [5]. We define HourglassStack in Table 4. After each stack, we use two separate branches for predicting heatmaps and human center depth. During training, we backpropagate gradient through all stacks, while during inference, we only use the predictions from final stack.

Index	Input	Data	Operator	Output shape
(1)	-	feat.	-	$256 \times 64 \times 112$
(2)	(1)		$\begin{bmatrix} 3 \times 3, \text{stride } 1 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$256 \times 64 \times 112$
(3)	(2)		$\begin{bmatrix} 3 \times 3, \text{stride } 2 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$384 \times 32 \times 56$
(4)	(3)		$\begin{bmatrix} 3 \times 3, \text{stride } 1 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$384 \times 32 \times 56$
(5)	(4)		$\begin{bmatrix} 3 \times 3, \text{stride } 2 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$512 \times 16 \times 28$
(6)	(5)		$\begin{bmatrix} 3 \times 3, \text{stride } 2 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$512 \times 8 \times 14$
(7)	(6)		$\begin{bmatrix} 3 \times 3, \text{stride } 1 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$512 \times 8 \times 14$
(8)	(7)		$\begin{bmatrix} 3 \times 3, \text{stride } 1 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$512 \times 8 \times 14$
(9)	(8)		Upsample $2\times$	$512 \times 16 \times 28$
(10)	(7), (9)		Sum	$512 \times 16 \times 28$
(11)	(10)		$\begin{bmatrix} 3 \times 3, \text{stride } 1 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$384 \times 16 \times 28$
(12)	(10)		Upsample $2\times$	$384 \times 32 \times 56$
(13)	(4), (12)		Sum	$384 \times 32 \times 56$
(14)	(13)		$\begin{bmatrix} 3 \times 3, \text{stride } 1 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$256 \times 32 \times 56$
(15)	(14)		Upsample $2\times$	$256 \times 64 \times 112$
(16)	(2), (15)	refined feat.	Sum	$256 \times 64 \times 112$

Table 4: **Modular architecture for one HourglassStack.** We follow the design and implementation of [4]. Each convolutional block denoted in the bracket has an internal skip connection with appropriate strides. We use nearest upsampling operator.

Index	Input	Data	Operator	Output shape
(1)	-	scene image	-	$3 \times 256 \times 448$
(2)	-	stacked heatmaps	-	$(N \times J) \times 64 \times 112$
(3)	(1)		7×7, stride 2	$64 \times 128 \times 224$
(4)	(3)		MaxPool, stride 2	$64 \times 64 \times 112$
(5)	(4)		$\begin{bmatrix} 3 \times 3, \text{stride } 1 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$64 \times 64 \times 112$
(6)	(2)		$\begin{bmatrix} 3 \times 3, \text{stride } 1 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$64 \times 64 \times 112$
(7)	(5), (6)		$\begin{bmatrix} 3 \times 3, \text{stride } 2 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$128 \times 32 \times 56$
(8)	(7)		$\begin{bmatrix} 3 \times 3, \text{stride } 2 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$256 \times 16 \times 28$
(9)	(8)		$\begin{bmatrix} 3 \times 3, \text{stride } 2 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$512 \times 8 \times 14$
(10)	(9)	encoder feat.	GlobalAvgPool	$512 \times 1 \times 1$
(11)	(10)	μ	Linear	$Z \times 1 \times 1$
(12)	(10)	σ	Linear	$Z \times 1 \times 1$
(13)	(11), (12)	$\mathbf{z}$	Sample from $\mathcal{N}(\mu, \sigma)$	$Z \times 8 \times 14$
(14)	(13)		3×3, stride 1	$512 \times 8 \times 14$
(15)	(14)		$\begin{bmatrix} 3 \times 3, \text{stride } 1 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$512 \times 8 \times 14$
(16)	(15)		Upsample $2\times$	$512 \times 16 \times 28$
(17)	(16)		$\begin{bmatrix} 3 \times 3, \text{stride } 1 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$256 \times 16 \times 28$
(18)	(17)		Upsample $2\times$	$256 \times 32 \times 56$
(19)	(18)		$\begin{bmatrix} 3 \times 3, \text{stride } 1 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$128 \times 32 \times 56$
(20)	(19)		Upsample $2\times$	$128 \times 64 \times 112$
(21)	(20)	decoder feat.	$\begin{bmatrix} 3 \times 3, \text{stride } 1 \\ 3 \times 3, \text{stride } 1 \end{bmatrix}$	$64 \times 64 \times 112$
(22)	(21)	goal heatmap pred.	1×1, stride 1	$1 \times 64 \times 112$

Table 5: **Overall architecture for our GoalNet.** Each convolutional block denoted in the bracket has an internal skip connection with appropriate strides. Each convolutional operator is followed by a batch normalization and ReLU layer, except the one before heatmap prediction. We denote N as input time frames, J as the number of human keypoints, Z as the dimension of latent space. We set $Z = 30$ in our experiments. We use nearest upsampling operator.

References

1. Hassan, M., Choutas, V., Tzionas, D., Black, M.J.: Resolving 3d human pose ambiguities with 3d scene constraints. In: ICCV (2019) 1
2. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments.

TPAMI (2013) 1, 2
3. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 2
4. Law, H., Teng, Y., Russakovsky, O., Deng, J.: Cornernet-lite: Efficient keypoint based object detection. arXiv preprint arXiv:1904.08900 (2019) 7
5. Moon, G., Chang, J.Y., Lee, K.M.: Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image. In: ICCV (2019) 6
6. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: NIPS workshop (2017) 2
7. Pavllo, D., Feichtenhofer, C., Grangier, D., Auli, M.: 3d human pose estimation in video with temporal convolutions and semi-supervised training. In: CVPR (2019) 4
8. Savva, M., Chang, A.X., Hanrahan, P., Fisher, M., Nießner, M.: PiGraphs: Learning Interaction Snapshots from Observations. TOG (2016) 1
9. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NIPS (2017) 6
10. Wei, M., Miaomiao, L., Mathieu, S., Hongdong, L.: Learning trajectory dependencies for human motion prediction. In: ICCV (2019) 4