ALIEN: Implicit Neural Representations for Human Motion Prediction under Arbitrary Latency
——Supplementary Material——

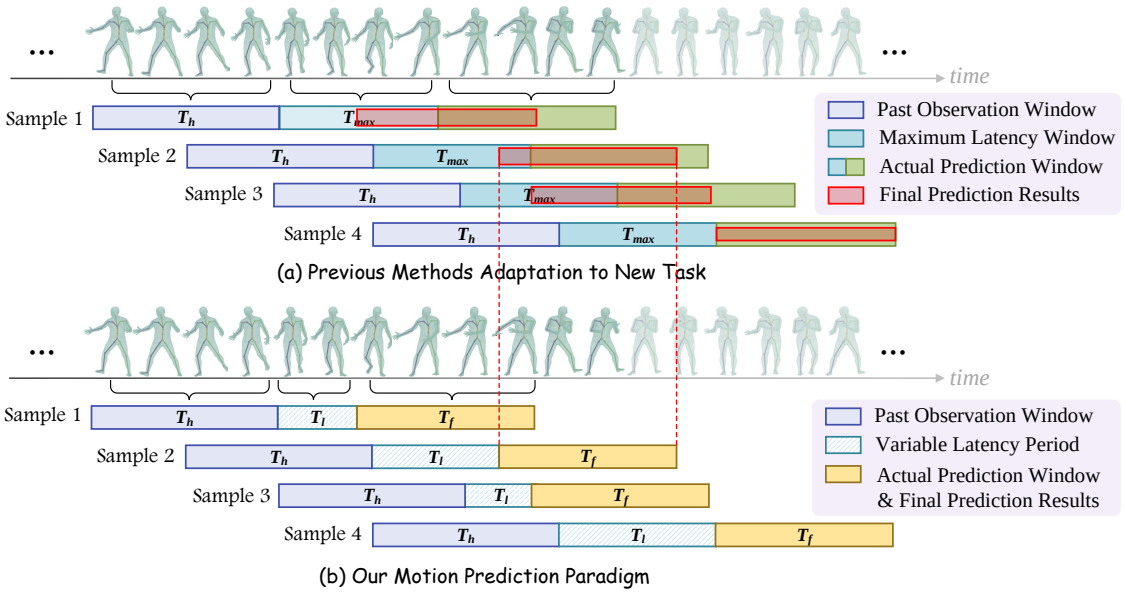


Figure 1. An illustration of (a) previous methods adaption for latency-aware motion prediction task, and (b) our proposed motion prediction paradigm for this task.

In the following, we first present the detailed process of the latency-aware motion prediction task in Section A and provide more implementation details in Section B. We then outline the training and inference algorithms of the proposed model in Section C. In Section D, we present and analyze further experimental results. Finally, we discuss the limitations of our approach and directions for future work in Section E.

A. Latency-Aware Motion Prediction

We illustrate previous baselines and our ALIEN to address the problem of predicting future poses while accounting for arbitrary latency in Figure 1. Given past T_h frames $\mathbf{X}_{1:T_h}$, our model will directly predict future $\mathbf{X}_{T_h+T_l+1:T_h+T_l+T_f}$ after latency period with variable length T_l. However, previous methods must predict poses over an extended window that contains both the maximum latency window (T_{max} frames) and the original prediction window (T_f frames). Then, they extract the relevant segment (T_f frames) as the final prediction results. Consequently, these methods can forcefully consider arbitrary latency, but requires additional effort to predict T_{max} poses unrelated to the original prediction window.

B. More Implementation Details

In this section, we further introduce more implementation details that are specific to each dataset. For Human3.6M dataset, we set the number of DCT coefficients as 10. For CMU-MoCap dataset, we set the number of DCT coefficients as 8. For 3DPW dataset, we set the number of DCT coefficients as 10. The learning rate is set to 1e-4 with a 0.96 decay every two epochs. The batch size is set to 16 and the gradients are clipped to a maximum l_2-norm of 1. We implement the network using Pytorch, and we use Adam to train this model for 150 epochs.

C. Training & Inference Algorithm

The training and inference algorithms of our proposed method ALIEN are reported in Algorithm 1 and 2, respectively. During training, the hyper-network parameter

Algorithm 1: ALIEN Training Procedure

Require: Learning rate α, trade-off parameter λ.
Output: Paramters $\Theta = \{\varphi, \mathbf{Z}, \phi_p, \phi_r\}$, including hyper-network parameter φ, base network $\mathbf{Z}$, and head parameter ϕ_p, ϕ_r.

while *not converged* **do**
- Sample a batch data from the training dataset $\mathcal{X}$: $\{\mathbf{X}^{(n)}_{1:T_h}, \mathbf{X}^{(n)}_{T_h+1:T_h+T_l}, \mathbf{X}^{(n)}_{T_h+T_l+1:T_h+T_l+T_f}\}_{n=1}^{N}$
- Obtain the instance-specific parameter: $\psi^{(n)} = \varphi(\mathbf{X}^{(n)}_{1:T_h})$
- Split instance-specific tokens into matrices: $\mathbf{U}^n_l, \mathbf{V}^n_l, \theta^{(n)}_{b_l} = Split(\psi^{(n)})$
- Obtain the l-th layer weight in INR decoder: $\theta^{(n)}_{W_l} = \sigma(\mathbf{U}^{(n)}_l \otimes \mathbf{V}^{(n)T}_l) \odot \mathbf{Z}_l$
- Compute the prediction loss: $\mathcal{L}_{pred} := \frac{1}{N \cdot T_f} \sum_{n=1}^{N} \sum_{t=T_h+T_l+1}^{T_h+T_l+T_f} \left\| \mathbf{X}^{(n)}_t - g_{\phi_p}(f_{\theta^{(n)}_p}(t)) \right\|_2^2$
- Compute the reconstruction loss: $\mathcal{L}_{rec} := \frac{1}{N \cdot T_h} \sum_{n=1}^{N} \sum_{t=T_l+1}^{T_h+T_l} \left\| \mathbf{X}^{(n)}_t - g_{\phi_r}(f_{\theta^{(n)}_r}(t)) \right\|_2^2$
- Minimize the loss function by gradient descent: $\Theta \leftarrow \Theta - \alpha \nabla_\Theta (\mathcal{L}_{pred} + \lambda \mathcal{L}_{rec})$

end

φ, base network parameter $\mathbf{Z}$, and head parameter ϕ_p, ϕ_r are optimized at the same time. At inference, we feed the past observations into the hyper-network and output the instance-specific parameter ψ, which are then served as the weights of INR decoder to predict future motions while accounting for arbitrary latency. Compared to meta-learning framework, our hyper-network is a simple feed-forward network and can more comprehensively learn the dependencies of spatial and temporal modeling.

D. Additional Experiments

D.1. Any-Time Pose Prediction

In contrast to traditional human motion prediction methods, ALIEN leverages the flexibility of continuous neural motion representations to enable anytime pose forecasting, allowing predictions at arbitrary future time steps. This capability is particularly beneficial for decision-making in human-machine interaction systems, such as autonomous sweeping robots [11], which demand rapid and precise pose predictions at specific moments. Figure 2 compares our approach with previous methods, which rely on interpolation over pre-defined timestamps to approximate anytime forecasting. The results highlight that our model offers a more practical and adaptable solution for real-world applications in human motion prediction.

Algorithm 2: ALIEN Inference Procedure

Require: Past motion $\mathbf{X}_{1:T_h}$, network parameter Θ.
Output: Future motion $\hat{\mathbf{X}}_{T_h+T_l+1:T_h+T_l+T_f}$.

Obtain the instance-specific parameter: $\psi = \varphi(\mathbf{X}_{1:T_h})$
Split instance-specific tokens into matrices: $\mathbf{U}_l, \mathbf{V}_l, \theta_{b_l} = Split(\psi)$
Obtain the l-th layer weight in INR decoder: $\theta_{W_l} = \sigma(\mathbf{U}_l \otimes \mathbf{V}^T_l) \odot \mathbf{Z}_l$
Encode coordinates $[T_h + T_l + 1 : T_h + T_l + T_f]$: $\eta_0 = \gamma(t) = [\sin(\pi t), \cos(\pi t), \cdots]$
while $l < L + 1$ **do**
- $\eta_l = \text{ReLU}(\theta^{(n)}_{W_l} \eta_{l-1} + \theta^{(n)}_{b_l})$

end
Compute the final prediction results: $\hat{\mathbf{X}}_{T_h+T_l+1:T_h+T_l+T_f} = \phi_p(\eta_L)$

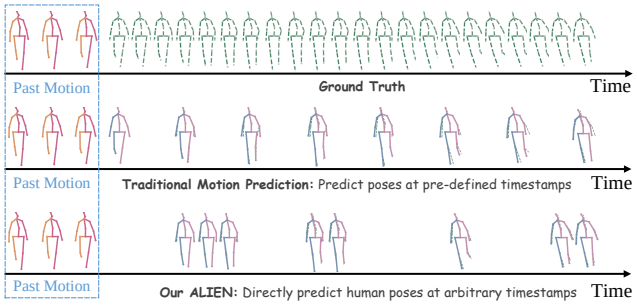


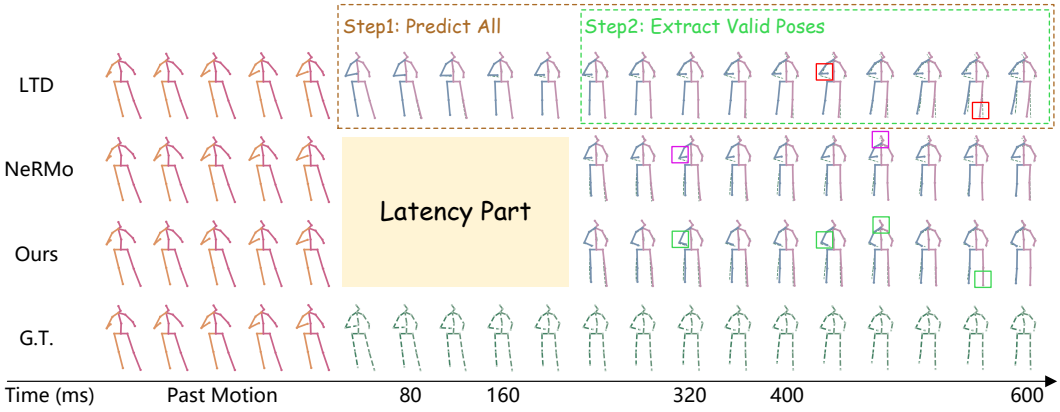
Figure 2. Our ALIEN supports flexible anytime pose forecasting, while previous methods can only predict human poses at fixed and pre-defined timestamps.

Action	Hammer	Lift	Prec.	Rnd.	Polishing	Heavy	Light	Avg.
LTD [6]	42.3	68.4	52.0	53.1	41.7	64.3	62.4	54.9
SeS-GCN [10]	41.1	61.9	**46.3**	48.7	38.6	56.5	56.9	50.0
Ours	**39.2**	**58.7**	46.9	**48.0**	**38.4**	**55.7**	**53.9**	**48.7**

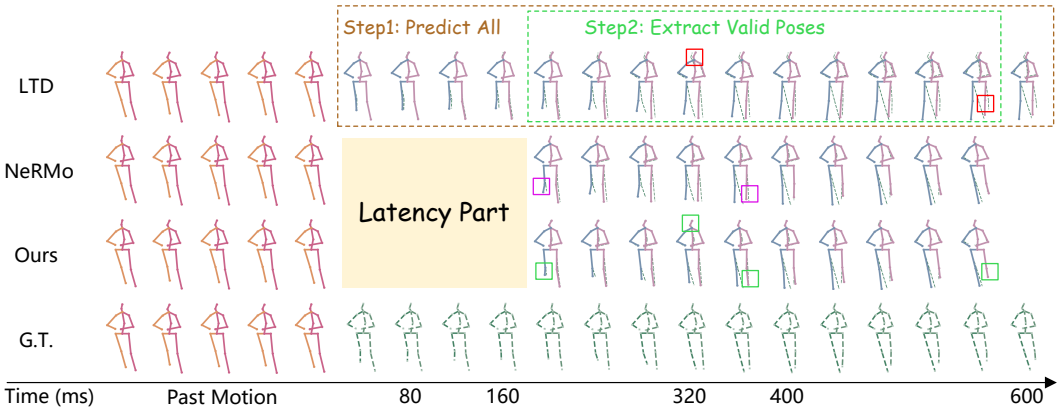
Table 1. Comparison results on CHICO dataset for LTD, SeS-GCN and the proposed method.

D.2. Results on CHICO Dataset

In this section, we provide additional experiments on Cobots and Humans in Industrial COllaboration (CHICO) dataset [10] that can better reflect the real-world scenarios. The dataset contains a single operator in a smart factory environment performing seven assembly tasks together with a Kuka LBR robot in a marker-less setup, including Lightweight pick and place, Heavyweight pick and place, surface polishing, precision pick and place, random pick and place, high shelf lifting and hammering. Accurate forecast with latency is useful for remote operators to anticipate collisions between human and robots. We follow settings as SeS-GCN [10], but consider arbitrary latency. The results



(a) Action: discussion, latency length: 5



(b) Action: walkingtogether, latency length: 4

Figure 3. Visualization comparison of baselines and our proposed method with different latency lengths: (a) "discussion" motion sample with $T_l = 5$, (b) "walkingtogether" motion sample with $T_l = 4$.

of each action at 400ms are reported in Table 1. We can observe that our proposed method outperforms baselines (LTD, SeS-GCN), showing its effectiveness in real-world applications.

D.3. More Visualization Results

In Figure 3, we provide additional qualitative results comparing our method with baselines under different latency lengths, T_l. Each sub-figure displays the results from top to bottom: LTD [6], NeRMo [12], our proposed ALIEN, and the ground truth. We can observe that our predictions consistently align more closely with the ground truth than other baselines, regardless of the latency duration. This demonstrates that our method is more robust in handling variable latency, producing more accurate motion predictions.

D.4. Ablation on INR Decoder Architecture

We investigate the impact of Fourier feature embedding, the number of linear layers and hidden dimension on prediction performance. Table 2 presents the prediction errors on the Human3.6M dataset, where "Num." denotes the number of linear layers, and "Dim." denotes the number of neurons of hidden layer. From the table, it is evident that the Fourier feature embedding of temporal coordinates is crucial for implicit neural representations. Additionally, a 5-layer MLP structure with 256 hidden dimension is adequate for capturing temporal information in our shared INR decoder.

Fourier	Num.	Dim.	80ms	160ms	320ms	400ms	600ms
✗	3	256	13.9	31.2	62.8	80.7	115.6
✗	5	256	13.2	29.5	61.3	78.4	112.0
✓	1	256	12.0	28.8	57.7	73.4	99.60
✓	3	256	9.9	23.2	50.2	63.8	85.1
✓	5	256	**9.7**	**21.8**	47.0	58.3	**79.3**
✓	7	256	**9.7**	21.9	**46.8**	58.4	79.6
✓	5	512	9.8	22.1	47.0	**58.2**	79.5

Table 2. Ablation study on INR decoder architecture designs.

D.5. More Results on CMU-MoCap

Table 3 provides the prediction performance of baselines and our method across 7 actions on CMU-MoCap dataset

scenarios	basketball				basketball signal				jumping				running			
millisecond	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms
Res-sup. [8]	15.45	26.88	43.51	49.23	20.17	32.98	42.75	44.65	26.85	48.07	93.50	108.90	25.76	48.91	88.19	100.80
LTD [6]	11.68	21.26	40.99	50.78	3.33	6.25	13.58	17.98	17.18	32.37	60.12	72.55	14.53	24.20	37.44	41.10
MSR-GCN [3]	10.28	18.94	37.68	47.03	3.03	5.68	12.35	16.26	14.99	28.66	55.86	69.05	12.84	20.42	30.58	34.42
PGBIG [5]	9.53	17.53	**35.32**	**44.23**	2.71	4.88	10.77	14.63	13.93	27.78	55.80	69.01	12.69	23.18	38.31	42.24
SPGSN [4]	10.24	18.54	38.22	48.68	2.91	5.25	11.31	15.01	14.93	28.16	56.72	71.16	10.75	16.67	26.07	30.08
NeRMo [12]	9.47	**16.87**	36.61	47.40	2.73	4.93	11.59	15.84	**13.07**	**25.65**	**54.92**	70.58	**9.72**	**15.31**	25.99	31.16
Ours	**9.39**	16.94	36.08	46.12	**2.70**	**4.82**	**10.66**	**14.50**	13.51	26.81	55.16	**68.48**	9.91	15.70	**25.53**	**29.81**
scenarios	soccer				walking				washing window				average			
millisecond	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms
Res-sup. [8]	17.75	31.30	52.55	61.40	44.35	76.66	126.83	151.43	22.84	44.71	86.78	104.68	24.74	44.21	76.30	88.73
LTD [6]	13.33	24.00	43.77	53.20	6.62	10.74	17.40	20.35	5.96	11.62	24.77	31.63	9.94	18.02	33.55	40.95
MSR-GCN [3]	10.92	19.50	37.05	46.38	6.31	10.30	17.64	21.12	5.49	11.07	25.05	32.51	8.72	15.83	30.57	38.10
PGBIG [5]	11.09	20.62	39.48	48.72	6.23	10.34	16.84	19.76	4.63	**9.16**	**20.87**	**27.34**	8.20	15.41	30.13	37.27
SPGSN [4]	10.86	18.99	**35.05**	**45.16**	6.32	10.21	16.34	20.19	**4.86**	9.44	21.50	28.37	8.30	14.80	28.64	36.96
NeRMo [12]	10.54	**18.03**	36.93	47.03	6.09	9.15	16.18	19.34	4.87	9.52	23.83	29.12	**8.05**	**14.14**	29.43	37.15
Ours	**10.23**	18.38	36.78	45.91	**5.99**	**9.07**	**15.38**	**18.89**	4.95	10.16	22.96	29.10	8.09	14.55	**28.62**	**36.10**

Table 3. Comparisons of different methods for 7 actions on CMU-Mocap dataset for conventional human motion prediction setting. The best results are highlighted in bold, and the second best are marked by underline.

for conventional zero-latency motion prediction setting. According to the table, we can see that our method outperforms other baselines in most cases, especially for predictions at 320ms and 400ms. The experimental results supplement to Section 4.4 of the main paper.

E. Limitation and Future Work

Our method in current version focuses on human motion modeling, and cannot accommodate variations in body shapes. Some recent works [1, 2, 9] have utilized implicit neural representations to model articulated objects with varying body structures. Combining our model with them could open up new possibilities, such as animating and rendering specific characters performing novel motions. Another limitation is that our model does not account for global translations during motion prediction. The absence of scene context may lead to artifacts such as "ghost motions" [7]. Therefore, it would be interesting to extend our model to contact-aware motion prediction, enabling more accurately handle global translations and interactions with the environment.

References

[1] Yukang Cao, Yan-Pei Cao, Kai Han, Ying Shan, and Kwan-Yee K Wong. Dreamavatar: Text-and-shape guided 3d human avatar generation via diffusion models. In *CVPR*, pages 958–968, 2024. 4

[2] Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. Snarf: Differentiable forward skinning for animating non-rigid neural implicit shapes. In *ICCV*, pages 11594–11604, 2021. 4

[3] Lingwei Dang, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction. In *ICCV*, pages 11467–11476, 2021. 4

[4] Maosen Li, Siheng Chen, Zijing Zhang, Lingxi Xie, Qi Tian, and Ya Zhang. Skeleton-parted graph scattering networks for 3d human motion prediction. In *ECCV*, pages 18–36, 2022. 4

[5] Tiezheng Ma, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. Progressively generating better initial guesses towards next stages for high-quality human motion prediction. In *CVPR*, pages 6437–6446, 2022. 4

[6] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *ICCV*, pages 9489–9497, 2019. 2, 3, 4

[7] Wei Mao, Richard I Hartley, Mathieu Salzmann, et al. Contact-aware human motion forecasting. *NeurIPS*, 35: 7356–7367, 2022. 4

[8] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *CVPR*, pages 2891–2900, 2017. 4

[9] Marko Mihajlovic, Shunsuke Saito, Aayush Bansal, Michael Zollhoefer, and Siyu Tang. Coap: Compositional articulated occupancy of people. In *CVPR*, pages 13201–13210, 2022. 4

[10] Alessio Sampieri, Guido Maria D'Amely di Melendugno, Andrea Avogaro, Federico Cunico, Francesco Setti, Geri Skenderi, Marco Cristani, and Fabio Galasso. Pose forecasting in industrial human-robot collaboration. In *ECCV*, pages 51–69, 2022. 2

[11] Zixing Wang and Ahmed H Qureshi. Anypose: Anytime 3d human pose forecasting via neural ordinary differential equations. *arXiv preprint arXiv:2309.04840*, 2023. 2

[12] Dong Wei, Huaijiang Sun, Xiaoning Sun, and Shengxiang Hu. Nermo: Learning implicit neural representations for 3d human motion prediction. In *ECCV*, pages 409–427, 2024. 3, 4