ReliaAvatar: A Robust Real-Time Avatar Animator with Integrated Motion Prediction

Bo Qian, Zhenhuan Wei, Jiashuo Li, Xing Wei*
School of Software Engineering, Xi'an Jiaotong University
{qb990531, zh-wei, xjtuljs}@stu.xjtu.edu.cn,weixing@mail.xjtu.edu.cn

Abstract

Efficiently estimating the full-body pose with minimal wearable devices presents a worthwhile research direction. Despite significant advancements in this field, most current research neglects to explore full-body avatar estimation under low-quality signal conditions, which is prevalent in practical usage. To bridge this gap, we summarize three scenarios that may be encountered in real-world applications: standard scenario, instantaneous data-loss scenario, and prolonged data-loss scenario, and propose a new evaluation benchmark. The solution we propose to address data-loss scenarios is integrating the full-body avatar pose estimation problem with motion prediction. Specifically, we present *ReliaAvatar*, a real-time, **relia**ble **avatar** animator equipped with predictive modeling capabilities employing a dual-path architecture. ReliaAvatar operates effectively, with an impressive performance rate of 109 frames per second (fps). Extensive comparative evaluations on widely recognized benchmark datasets demonstrate ReliaAvatar's superior performance in both standard and low data-quality conditions. The code is available at https://github.com/MIV-XJTU/ReliaAvatar.

1 Introduction

Virtual reality, augmented reality (AR), and mixed reality (MR) are rapidly evolving fields that offer new dimensions for human communication and interaction. A critical aspect of enhancing these immersive experiences lies in the ability of models to generate seamless and realistic avatar motions, ideally using user-friendly devices such as head-mounted displays (HMD) or non-wearable WiFi devices [Yan *et al.*, 2024]. Despite notable advancements in using sparse observations to animate full-body avatars, a significant research gap exists to effectively drive avatars in scenarios plagued by low-quality signals.

In practical applications, poor-quality signals are a common challenge, particularly when wearable devices are intended to be as convenient as possible. Various factors, including network fluctuations, occlusion in motion capture systems, and limited visibility of interactive handles in HMDs, can degrade signal integrity. Previous works such as AGRoL [Du *et al.*, 2023] and HMD-NeMo [Aliakbarian *et al.*, 2023] have addressed some aspects of these challenges. However, systematic exploration and comprehensive solutions for diverse data-loss scenarios remain underdeveloped.

In this work, we delve into potential data-loss scenarios that could occur in real-world applications. We specifically identify and focus on two scenarios: instantaneous data-loss and prolonged data-loss scenarios. Our response to these challenges is two-fold, encompassing both model architecture and training methodology innovations. We introduce ***Relia-***

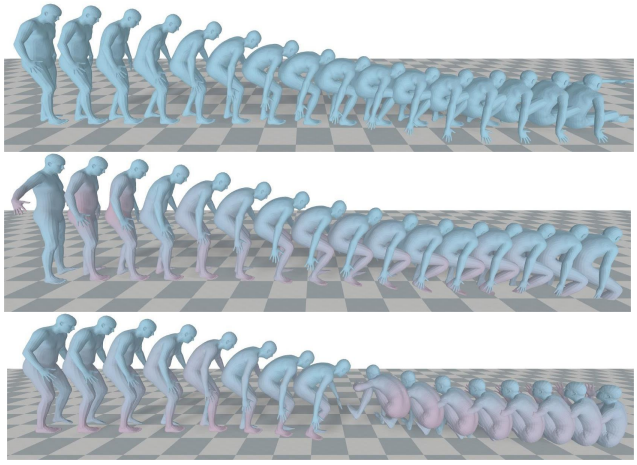


Figure 1: Visualization in the context of prolonged data-loss scenario. We mask out the latter half of a sample, which consisted of 80 frames and depicted "crouching". The first row represents the ground truth, the second row represents the response of ReliaAvatar, and the third row represents the response of AvatarPoser. The visualization clearly indicates that ReliaAvatar can operate effectively in prolonged data loss scenarios with only minor distortions. In contrast, AvatarPoser completely fails to perform in this scenario.

Method	Standard	Instantaneous	Prolonged
Other Methods	✓	×	×
ReliaAvatar	✓	✓	✓

Table 1: Ability to handle different scenarios.

*Corresponding author: Xing Wei.

Avatar, a real-time, robust, autoregressive Avatar Animator, integrating full-body joint motion prediction with an innovative training approach that simulates data-loss conditions.

We develop two distinct data-loss scenarios: **instantaneous**, resembling tracker signal loss like network packet loss, and **prolonged**, simulating long-last loss of specific motion capture joints, such as a hand-held controller's prolonged invisibility. Our model operates through two pathways: a regression pathway for conventional full-body avatar pose estimation tasks and a prediction pathway that predicts motions in the absence of tracker signals, ensuring continuity in avatar movements. Both pathways leverage GRU-based models for feature extraction. The combined features are then transformed into a decoding token sequence representing 22 SMPL joints, which will be used in a Transformer encoder to model the inter-joint relationships. We also propose an autoregressive training pipeline encompassing three preprocessing methods aligned with standard, instantaneous, and prolonged data-loss scenarios. During training, each signal sequence is processed using one of these preprocessing methods, enabling the model to adapt to the challenges posed by different data loss scenarios.

We compare ReliaAvatar with existing methods in all scenarios on the AMASS benchmark dataset [Mahmood *et al.*, 2019]. The results showcase our model's state-of-the-art performance in both standard and data-loss scenarios. We also provide a comparison with AvatarPoser [Jiang *et al.*, 2022] in Figure 1 in the prolonged data-loss scenario. These qualitative and quantitative results collectively demonstrate that we are the first method, as stated in Table 1, that has robustness to data quality. Our model also demonstrates impressive performance during the online inference stage, achieving a remarkable 109 fps. This outstanding speed surpasses other avatar pose estimation methods, highlighting the superiority of our model in real-time applications.

Our contribution can be summarized as follows:

- We pioneer a comprehensive investigation into practical data-loss scenarios, identifying and focusing on two key scenarios: instantaneous and prolonged data loss.
- We propose ReliaAvatar, a real-time, robust avatar animator with integrated full-body joint motion prediction, and an autoregressive training pipeline tailored for data-loss scenarios. This approach significantly enhances the model's adaptability and performance.
- Our experimental results demonstrate that ReliaAvatar not only achieves top-tier performance in the standard scenario but also represents, to our knowledge, the first method adept at managing various data-loss scenarios effectively. Furthermore, ReliaAvatar outperforms other methods in terms of computational efficiency.

2 Related Work

2.1 Full-body Avatar Pose Estimation

In the past, motion capture systems required users to wear numerous devices, as seen in early implementations [Xsens, 2013; Vlasic *et al.*, 2007]. However, this requirement often compromises the immersive experience, highlighting the need for more compact approaches. Consequently, the focus shifted to generating a full-body pose using sparse observations, a topic that has garnered significant attention and led to numerous advancements [Von Marcard *et al.*, 2017; Huang *et al.*, 2018; Yi *et al.*, 2021; Yang *et al.*, 2021; Jiang *et al.*, 2022; Zheng *et al.*, 2023; Du *et al.*, 2023; Aliakbarian *et al.*, 2023]. These innovative methods allow users to wear tracking devices on only a limited number of joints, such as the head, hands, and pelvis, thus improving the overall user experience. Among these developments, AvatarPoser [Jiang *et al.*, 2022] introduced a Transformer-based architecture combined with an inverse kinematics (IK) module, paving the way for more sophisticated avatar control techniques. AGRoL [Du *et al.*, 2023] utilized diffusion models to drive avatars, showing notable robustness against instantaneous data loss. Similarly, HMD-NeMo [Aliakbarian *et al.*, 2023] made strides in avatar control by focusing on scenarios where the hands are partially or entirely obscured. Building on these foundational works, our research delves deeper into the realm of full-body avatar estimation under the conditions of low-quality data.

2.2 3D Human Motion Prediction

A key innovation of our model is the motion prediction pathway in the architecture, which is used to provide additional cues to avatar pose estimation. There are many motion prediction methods available for reference, ranging from nonlinear Markov models [Lehrmann *et al.*, 2014] and Restricted Boltzmann Machines [Taylor *et al.*, 2006], to more recent developments like Graph Convolutional Networks [Ma *et al.*, 2022; Mao *et al.*, 2020; Mao *et al.*, 2019] and Recurrent Neural Networks [Jain *et al.*, 2016; Liu *et al.*, 2019; Martinez *et al.*, 2017]. In the prediction pathway, we use a GRU module[Cho *et al.*, 2014] to extract features and a Transformer [Vaswani *et al.*, 2017] module to model the inter-joint relationships. The prediction pathway can be aggregated with the regression pathway at the joint-relation Transformer to form our reliable avatar animator.

3 Method

3.1 Problem Definition

Task. The core objective of our task is to accurately predict the 3D full-body human poses y^t at any given time-step t, using sparse joint signals x^t and supplementary information e^t. This supplementary information, e^t, may include data derived from historical movements or other available motion signals at time-step t. Therefore, our objective is formally defined as follows.

$$\max_{\theta} \mathbb{P}(y^t | x^t, e^t, f_\theta) \tag{1}$$

Traditionally, this task is approached as an upsampling challenge, where the goal is to learn the mapping from sparse joint signals to a complete full-body joint configuration. In these conventional methods, inputs are configured as a series of continuous sparse signals over a time window to facilitate the extraction of temporal features, leading to the following model formulation.

$$\hat{y}^t = f_\theta(x^t, x^{t-1}, ..., x^{t-T}) \tag{2}$$

Here, the supplementary information e^t is represented as $e^t = \{x_i\}_{i=t-T}^{t-1}$, with T denoting the window size.

Those approaches neglect the continuity of the prediction process, which has been proven to be crucial in visual tracking[Wei *et al.*, 2023; Bai *et al.*, 2024]. Unlike them, we treat the task as an integrative process of both upsampling and full-body joint motion prediction. To this end, we incorporate historical trajectory states as an additional input variable. We utilize a Gated Recurrent Unit (GRU) to extract temporal features from both the tracker signals and the historical trajectory states. This approach allows us to redefine e^t as a composite of previous pose predictions and historical hidden states, i.e., $e^t = \{\hat{y}^{t-1}, h_x^{t-1}, h_y^{t-1}\}$. Consequently, our model's formulation is as follows:

$$\hat{y}^t, h_x^t, h_y^t = f_\theta(x^t, \hat{y}^{t-1}, h_x^{t-1}, h_y^{t-1}) \tag{3}$$

Here, h_x^t, h_y^t represent the hidden states generated by the GRU, essential for capturing the complex temporal dynamics of human motion.

Representation. ReliaAvatar's input signals consist of four parts: the tracker signals x^t, hidden tracker states h_x^{t-1}, the historical trajectory states $\hat{y}^{t-1}$, and hidden historical states h_y^{t-1}. Here, $\hat{y}^{t-1}$ represents the output of the model at time-step $t-1$, and h_x^{t-1} and h_y^{t-1} are the hidden states returned by the GRU module.

The input tracker signals x^t at time-step t are composed of the orientation[1] $x_r \in \mathbb{R}^{|\mathbb{J}|\times 6}$, rotation velocity $\Delta x_r \in \mathbb{R}^{|\mathbb{J}|\times 6}$, position $x_p \in \mathbb{R}^{|\mathbb{J}|\times 3}$, and linear velocity $\Delta x_p \in \mathbb{R}^{|\mathbb{J}|\times 3}$ of the trackers. $\mathbb{J}$ represents the set of all trackers, $\Delta x_p = x_p^t - x_p^{t-1}$, $\Delta x_r = f_{6D}(R^{-1}(x_r^{t-1})R(x_r^t))$, and R and R^{-1}, respectively, represent converting the 6D representation of rotation into the corresponding rotation matrix and inverse matrix, where f_{6D} represents converting the rotation matrix into a 6D representation. Therefore, $x^t = \{x_r^t, \Delta x_r^t, x_p^t, \Delta x_p^t\} \in \mathbb{R}^{|\mathbb{J}|\times 18}$.

The historical trajectory states $\hat{y}^t$ at time $t+1$ are also composed of the four parts: 22 SMPL joints' local rotation relative to their parent joints $\hat{y}_r^t \in \mathbb{R}^{22\times 6}$, position $\hat{y}_p^t \in \mathbb{R}^{22\times 3}$ and corresponding velocity $\Delta\hat{y}_r^t \in \mathbb{R}^{22\times 6}$ and $\Delta\hat{y}_p^t \in \mathbb{R}^{22\times 3}$. Therefore, the overall output can be represented as $\hat{y}^t = \{\hat{y}_r^t, \Delta\hat{y}_r^t, \hat{y}_p^t, \Delta\hat{y}_p^t\} \in \mathbb{R}^{396}$.

Scenarios. Current models often operate under the assumption that tracker signals are consistently available, promptly received, and accurate. However, real-world applications present more complex and varied scenarios in which the quality and consistency of tracker signals cannot always be guaranteed. To address this, we have identified three primary scenarios in which our model and similar models may need to operate:

- **Standard scenario:** This scenario represents ideal conditions where the model successfully receives all required tracker signals without any loss or degradation. This is the standard operational scenario for most existing methods.
- **Instantaneous data-loss scenario:** Real-world applications are prone to transient disruptions, such as network fluctuations, leading to the momentary loss of specific tracker signals. To simulate these sudden and brief interruptions in signal transmission in this scenario, we make each tracker signal at any given time t subject to a probability p of being lost, as illustrated in Fig 3.A.
- **Prolonged data-loss scenario:** There are instances where certain tracker signals may be consistently unavailable over a long-lasting period[2]. Such situations can arise from continuous occlusion of infrared markers or sustained invisibility of hand-held controller signals in an HMD kit. This scenario is characterized by a continuous loss of all tracker signals for a specific duration, as depicted in Fig 3.B. It represents challenges posed by prolonged data unavailability.

The detailed configurations and implications of these scenarios will be further explored in Section 3.4 for the training stage and Section 4.3 for the testing stage, respectively.

3.2 Overview

ReliaAvatar adopts an autoregressive pipeline during both the training and testing stages. The process of ReliaAvatar's handling continuous tracker signal sequences can be represented by the following equations:

$$\hat{y}^0, h_x^0, h_y^0 = f_\theta(x^0) \tag{4}$$

$$\hat{y}^1, h_x^1, h_y^1 = f_\theta(x^1, \hat{y}^0, h_x^0, h_y^0) \tag{5}$$

$$\ldots$$

$$\hat{y}^t, h_x^t, h_y^t = f_\theta(x^t, \hat{y}^{t-1}, h_x^{t-1}, h_y^{t-1}) \tag{6}$$

$$\ldots$$

During the training stage, given a tracker signal sequence $\{x^t\}_{t=0}^{L-1}$ as input, the training process is divided into L sequential steps with each step building upon the previous one. L is the length of the sequence.

As depicted in Figure 2, our model features two distinct yet interconnected pathways: the regression pathway (Regression Encoder→Joint-relation Transformer→Decoder) and the prediction pathway (Prediction Encoder→Joint-relation Transformer→Decoder). The regression pathway operates similarly to the traditional avatar animator. Its primary function is to regress full-body poses from sparse joint information. This process can be formulated as:

$$\hat{y}^t, h_x^t = f_\theta^{reg}(x^t, h_x^{t-1}) \tag{7}$$

The prediction pathway, on the other hand, aligns more closely with motion prediction models. It utilizes known sequences of past motion to predict the current full-body motion. The formulation for this pathway is as follows:

$$\hat{y}^t, h_y^t = f_\theta^{pred}(\hat{y}^{t-1}, h_y^{t-1}) \tag{8}$$

Both pathways converge at the Joint-Relation Transformer. This convergence results in the complete model represented

[1]The orientation is represented by the 6D representation of rotation [Zhou *et al.*, 2019].

[2]The term "long-lasting period" refers to a period that is significantly longer than the frame interval, for example, 1 second.

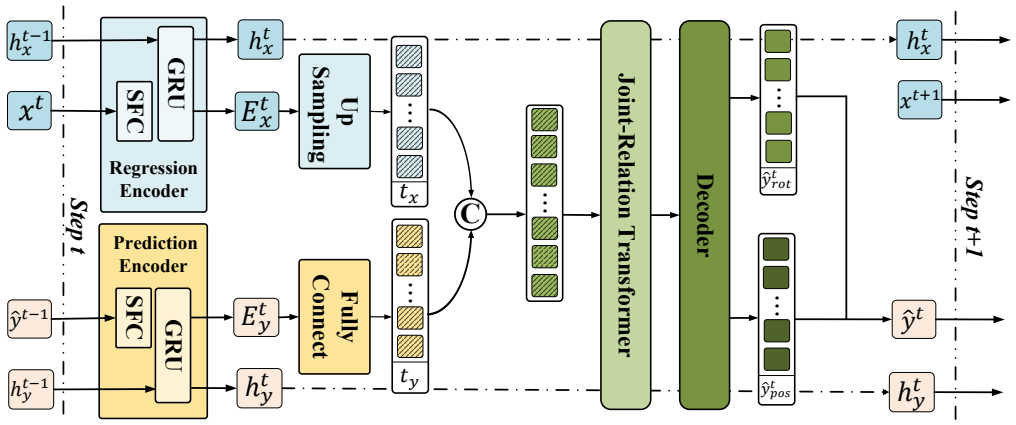
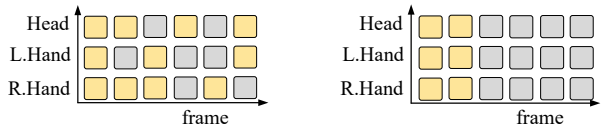


Figure 2: Illustration of our dual-pathway, autoregressive framework. ReliaAvatar has two pathways: the regression pathway (Regression Encoder→Joint-Relation Transformer→Decoder) and the prediction pathway(Prediction Encoder→Joint-Relation Transformer→Decoder). The output of ReliaAvatar at time-step t forms a part of the input at time-step $t+1$. The blocks with diagonal lines as foreground (, ,) represent tokens that signify SMPL joints, e.g., pelvis, left wrist, right ankle.



A) Instantaneous data-loss **B)** Prolonged data-loss

Figure 3: Illustration of data-loss scenarios. represents that the signals have been received properly, while represents that the signals are lost.

by Eq. (3). The Joint-Relation Transformer and the decoder are shared modules utilized by both pathways, while each pathway has its unique components – the regression encoder for the regression pathway and the prediction encoder for the prediction pathway.

3.3 Model Architecture

Regression Encoder and Prediction Encoder. As shown in Figure 2, tracker signals are initially processed through a sparse fully connected layer (SFC) to extract initial features $E_0^t \in \mathbb{R}^{|\mathbb{J}|\times d}$, where d is the embedding dimension of the model. The purpose of SFC is to merge multiple linear layers into a sparse linear layer, thereby reducing the number of loops in the forward process and improving the runtime efficiency. In the regression encoder, SFC is formed by merging four linear layers that handle orientation x_r, rotation velocity Δx_r, position x_p, and linear velocity Δx_p. This module can be formulated as:

$$E_0 = \begin{bmatrix} x_r \\ \Delta x_r \\ x_p \\ \Delta x_p \end{bmatrix} \begin{bmatrix} W_1^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & W_2^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & W_3^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & W_4^T \end{bmatrix} + b \qquad (9)$$

Then, the GRU module will extract the temporal features, as described in the following formula:

$$E_x^t, h_x^t = GRU(E_0^t, h_x^{t-1}) \qquad (10)$$

After the regression encoder, the features of all tracker signals $E_x^t \in \mathbb{R}^{|\mathbb{J}|\times d}$ are obtained. Subsequently, an upsampling module will be used to increase the dimension to $22 \times d$, thus obtaining the decoding token sequence $\{t_x^i\}_{i=0}^{21}$ which will be input to the Joint-Relation Transformer. Each token represents a SMPL joint.

The prediction encoder is similar to the regression encoder. After passing through an SFC and a GRU module, $\hat{y}^{t-1}$ is mapped to a token $E_y^t \in \mathbb{R}^d$. Then a fully connected layer is applied to expand it to 22 joint tokens $\{t_y^i\}_{i=0}^{21}$.

Joint-Relation Transformer. In human kinematics, the motions of the entire body are not simply a combination of independent motions of multiple joints. In contrast, the actions performed by the human body are strongly correlated with the relationship between joints. Therefore, it is evident that equipping the model with the ability to model joint relationships can enhance the driving effectiveness. A Transformer encoder is used to model joint relationships, which has been proven to be effective in PETR [Shi *et al.*, 2022].

The input of the Transformer is a token sequence $\{t^i\}_{i=0}^{21} = \text{Concat}\,(t_x, t_y) \in \mathbb{R}^{22\times(2d)}$ representing 22 SMPL joints, which is fused from the tokens generated by the previous two encoders.

Decoder. The decoder is responsible for decoding rotation $\hat{y}_{rot}^t$ and position $\hat{y}_{pos}^t$ directly from the output of the token sequence of the Transformer. The decoder is composed of two layers of MLP. There are three decoder variants:

- **Shared Decoder.** Since the pelvic joint parameter represents the global orientation of the entire body, while the other joint parameters represent local rotations relative to their parent joints, a dedicated decoder is used for the pelvic joint, while the remaining joints share a common decoder.
- **Multi-FC Decoder.** Each joint has its own dedicated decoder. During the inference process, the decoding of each joint is done sequentially.

- **SFC Decoder.** Use an MLP composed of SFCs as a unique decoder. As a result, all joints can be decoded simultaneously in parallel.

We adopted SFC decoder due to its advantages of high accuracy and high efficiency. The comparison of the three decoders will be mentioned in Sec 4.5.

3.4 Simulation Training

We simulate the three scenarios mentioned in Sec during training ReliaAvatar.

- **Standard scenario.** In the standard scenario, all tracker signals can be fully and correctly transmitted to the model. So in this scenario, there is no need for any masking of the signals.
- **Instantaneous data-loss scenario.** In the instantaneous data-loss scenario, we use a dropout layer to randomly mask the signals of each tracker in each frame with a probability of 0.1.
- **Prolonged data-loss scenario.** In the prolonged data-loss scenario, we mask out the latter half of each input sequence $\{x_t\}_{t=0}^{L-1}$. That is to say, $\{x_t\}_{t=L/2}^{L-1} = \mathbf{0}$.

Each sequence has an equal probability $(1/3)$ to undergo the above three simulation treatments.

3.5 Loss Function

We adopt orientation loss $\mathscr{L}_{ori}$, rotation loss $\mathscr{L}_{rot}$, SMPL position loss $\mathscr{L}_{pos}^{SMPL}$, decoded position loss $\mathscr{L}_{pos}^{dec}$, and velocity loss $\mathscr{L}_{vec}$ during the optimization process. All losses are computed using the L1 loss. Taking the rotation loss as an example, its loss function is as follows:

$$\mathscr{L}_{rot} = \frac{\Sigma_{t=0}^{L-1} ||\hat{y}_{rot}^t - y_{rot}^t||}{L} \tag{11}$$

Here, $\hat{y}_{rot}, y_{rot}$ represent the model output and corresponding ground truth, respectively. Other losses are computed in the same way. The whole loss function can be formulated as:

$$\begin{aligned}\mathscr{L} =& \lambda_{ori}\mathscr{L}_{ori} + \lambda_{rot}\mathscr{L}_{rot} + \lambda_{pos}^{SMPL}\mathscr{L}_{pos}^{SMPL} + \\ & \lambda_{pos}^{dec}\mathscr{L}_{pos}^{dec} + \lambda_{vec}L_{vec}\end{aligned} \tag{12}$$

Here, $\lambda_{ori}, \lambda_{rot}, \lambda_{pos}^{SMPL}, \lambda_{pos}^{dec}, \lambda_{vec}$ are weights of corresponding losses. The difference between $\mathscr{L}_{pos}^{SMPL}$ and $\mathscr{L}_{pos}^{dec}$ is that $\mathscr{L}_{pos}^{SMPL}$ is computed based on the position $SMPL(\hat{y}_{rot})$ inferred by SMPL and the ground truth y_{pos}, while $\mathscr{L}_{pos}^{dec}$ is computed based on the position $\hat{y}_{pos}$ decoded by the decoder and y_{pos}.

4 Experiments

4.1 Implementation Details

For all GRUs in the model, the number of layers is set to 1 and the dimension is set to 256. The joint-relation Transformer has 4 layers with a dimension of 512. The initial learning rate is set to 5×10^{-4}, and is halved every 15000 iterations. The length of the input sequence L is set to 32. $\{\lambda_{ori}, \lambda_{rot}, \lambda_{pos}^{SMPL}, \lambda_{pos}^{dec}, \lambda_{vec}\} = \{0.02, 1, 1, 1, 0.5\}$. The model is trained on two GeForce RTX 3090 GPUs for a total of 90000 iterations, with a batch size of 32 on each GPU. The training process costs approximately 32 hours to complete.

4.2 Evaluation Metrics

We adopt the following evaluation metrics.

- **MPJPE**: Mean Per Joint Position Error [cm].
- **MPJRE**: Mean Per Joint Rotational Error [degree].
- **MPJVE**: Mean Per Joint Velocity Error [cm/s].
- **fps**: Frames Per Second [frame].

4.3 Evaluation Protocols

Standard scenario. Following the experimental setup outlined in AvatarPoser [Jiang *et al.*, 2022], we partition the CMU [graphics lab., 2000], BMLrub [Troje, 2002], and HDM05 [Müller *et al.*, 2007] subsets into 90% training data and 10% testing data. For comparison, we evaluate the results using both three inputs (head and wrists) and four inputs (head, wrists, and pelvis). It is worth noting that, unless explicitly stated otherwise, all other results are based on the usage of three inputs.

Instantaneous data-loss scenario. AGRoL [Du *et al.*, 2023] proposed a setting for instantaneous data-loss, which randomly masks out 10% of the input sequence. However, in practical applications, an instantaneous data-loss does not necessarily mean the complete loss of all tracker signals. Instead, it might indicate the loss of information from a specific joint. Therefore, we propose a new setting where signals of each tracker signal have a certain probability p of being lost at each moment. We evaluated each model five times at $p = 0.1, 0.5$ and 0.9, and took the average as the result.

Prolonged data-loss scenario. In the scenario of prolonged data-loss, data is continuously lost for a long-last period. For evaluation purposes, we set a protocol to mask out subsequent M frames every 80 frames. We evaluated each model in $M = 20, 40$ and 60.

4.4 Comparison to the State-of-the-art

The evaluation results of each model in the standard scenario are extracted from the reports in their respective papers. For comparison in data-loss scenarios, we select several publicly available state-of-the-art methods (AvatarPoser [Jiang *et al.*, 2022], AGRoL [Du *et al.*, 2023] and AvatarJLM [Zheng *et al.*, 2023]). We conduct a fair comparison by retraining their publicly available code and evaluating them under the protocols for data-loss scenarios using the retrained parameters.

Standard scenario. To verify the effectiveness of our model, we perform a fair comparison with the state-of-the-art methods in the standard scenario. The results, as shown in Table 2 and Table 3, demonstrate that our model achieves state-of-the-art performance in three- and four-input conditions. This indicates that ReliaAvatar, despite being primarily designed to tackle data-loss scenarios, surpasses other models even in the standard scenario.

Instantaneous data-loss scenario. We compare ReliaAvatar with other models under three conditions: $p = 0.1, p = 0.5$, and $p = 0.9$. As shown in Table 4, ReliaAvatar demonstrates its superior robustness compared to other models in the instantaneous data-loss scenario. Under the conditions of

Method	MPJRE	MPJPE	MPJVE
FinalIK [Unity, 2018]	16.77	18.09	59.24
CoolMoves [Ahuja *et al.*, 2021]	5.20	7.83	100.54
LoBSTr [Yang *et al.*, 2021]	10.69	9.02	44.97
VAE-HMD [Dittadi *et al.*, 2021]	4.11	6.83	37.99
AvatarPoser [Jiang *et al.*, 2022]	3.21	4.18	29.40
EgoPoser [Jiang *et al.*, 2023]	-	4.14	25.95
AGRoL [Du *et al.*, 2023]	2.66	3.71	18.59
DAP [Di *et al.*, 2023]	2.69	3.68	24.30
AvatarJLM [Zheng *et al.*, 2023]	2.90	3.35	20.79
ReliaAvatar (Ours)	**2.53**	**3.18**	**18.30**

Table 2: State-of-the-art comparison on AMASS [Mahmood *et al.*, 2019] in the standard scenario using three inputs (*head and wrists*). Best in **bold**.

Method	MPJRE	MPJPE	MPJVE
FinalIK [Unity, 2018]	12.39	9.54	36.73
CoolMoves [Ahuja *et al.*, 2021]	4.58	5.55	65.28
LoBSTr [Yang *et al.*, 2021]	8.09	5.56	30.12
VAE-HMD [Dittadi *et al.*, 2021]	3.12	3.51	28.23
AvatarPoser [Jiang *et al.*, 2022]	2.59	2.61	22.16
AvatarJLM [Zheng *et al.*, 2023]	2.40	2.09	17.82
ReliaAvatar (Ours)	**2.16**	**1.94**	**14.32**

Table 3: State-of-the-art comparison on AMASS [Mahmood *et al.*, 2019] in the standard scenario using four inputs (*head, wrists, and pelvis*. Best in **bold**.

$p = 0.1$ and 0.5, ReliaAvatar is hardly affected. Even when the signals of each tracker have only a 10% probability of being received, ReliaAvatar can still operate normally.

Prolonged data-loss scenario. We compare our model with the previous state-of-the-art models in the prolonged data-loss scenario under three conditions: $M = 20, 40$, and 60. Table 5 indicates that existing models are unable to handle scenarios where there is a continuous loss of more than 40 frames of signals. In contrast, ReliaAvatar exhibits superior robustness compared to other methods, as it can function nearly normally even with a continuous loss of 20 frames. It is capable of operating even when the signal is continuously lost for 60 frames.

p	Method	MPJRE	MPJPE	MPJVE
0.1	AvatarPoser	9.01	20.02	1532.62
	AGRoL	<u>6.59</u>	<u>12.31</u>	<u>101.71</u>
	AvatarJLM	6.65	13.40	887.24
	ReliaAvatar(Ours)	**2.66**	**3.32**	**20.44**
0.5	AvatarPoser	15.47	53.07	3651.32
	AGRoL	<u>9.96</u>	<u>20.04</u>	<u>114.12</u>
	AvatarJLM	16.86	43.02	2512.13
	ReliaAvatar(Ours)	**2.77**	**3.51**	**26.51**
0.9	AvatarPoser	17.98	63.53	2582.13
	AGRoL	<u>13.08</u>	<u>27.59</u>	<u>117.26</u>
	AvatarJLM	25.16	65.50	1154.16
	ReliaAvatar(Ours)	**5.12**	**7.86**	**45.00**

Table 4: Comparison in instantaneous data-loss scenario. Best in **bold** and second best <u>underlined</u>.

M	Method	MPJRE	MPJPE	MPJVE
20	AvatarPoser	7.43	14.46	102.18
	AGRoL	<u>6.59</u>	<u>12.31</u>	<u>101.71</u>
	AvatarJLM	8.96	18.71	133.92
	ReliaAvatar (Ours)	**2.84**	**3.67**	**24.69**
40	AvatarPoser	11.52	23.57	123.33
	AGRoL	<u>9.96</u>	<u>20.24</u>	<u>114.12</u>
	AvatarJLM	14.52	34.39	154.17
	ReliaAvatar (Ours)	**3.48**	**5.17**	**34.01**
60	AvatarPoser	15.05	31.36	118.73
	AGRoL	<u>13.08</u>	<u>27.59</u>	<u>117.26</u>
	AvatarJLM	20.14	49.95	181.36
	ReliaAvatar (Ours)	**4.70**	**7.69**	**45.02**

Table 5: Comparison in prolonged data-loss scenario. Best in **bold** and second best <u>underlined</u>.

Input signals	MPJRE	MPJPE	MPJVE
$\{X\}$	2.69	3.49	23.49
$\{X, y_{rot}\}$	2.54	3.25	19.44
$\{X, y_{pos}\}$	2.58	3.28	18.39
$\{X, y_{rot}, y_{pos}\}$	2.53	3.25	19.07
$\{X, y_{rot}, y_{pos}, \Delta y_{rot}, \Delta y_{pos}\}$	**2.53**	**3.18**	**18.30**

Table 6: Ablation experiments of the input signals. Best in **bold**.

ReliaAvatar's robustness to low-quality signals stems from three aspects: 1) Our model abandons windowed inputs. Models that solely rely on a window of tracker signals as input (e.g., AvatarPoser with a window size of 40) exhibit obvious malfunctions when facing prolonged data-loss since the signals within the window are all set to zero. 2) Our model integrates motion prediction into the avatar animator. When the regression pathway fails to function properly, the prediction pathway can still predict the current full-body motion based on the historical trajectory states. 3) The autoregressive training paradigm allows for simulating abnormal scenarios that may occur in applications.

4.5 Ablation Studies

To illustrate the roles of various designs in our model, we conduct the following ablation experiments.

Input Signals. As shown in Table 6, we validate the performance gain of incorporating historical trajectory states into the input. Here, X represents the input used by most current methods, i.e. $X = \{x_r, \Delta x_r, x_p, \Delta x_p\}$.

If the historical trajectory states are not used as input, the regression pathway remains inactive. The experimental results demonstrate that incorporating $y_{rot}, \Delta y_{rot}, y_{pos}, \Delta y_{pos}$ into the input leads to performance gains. Due to the continuity of motion, the current full-body is naturally influenced by and aligned with the previous trajectories. So this improvement can be attributed to the previous trajectories providing cues for the generation of the current full-body motion.

Decoder. As shown in Table 7, we compare the performance of different decoder designs in our model. The model using a

Decoder	MPJRE	MPJPE	MPJVE	fps
Shared	2.64	3.38	19.51	**112.22**
Multi-FC	**2.51**	3.21	18.64	46.73
Sparse-FC	2.53	**3.18**	**18.30**	109.65

Table 7: Ablation experiments of the design of decoder. Best in **bold** and second best underlined.

method	MPJRE	MPJPE	MPJVE
add	2.66	3.31	19.70
concat (joint-dim)	2.66	3.31	19.83
concat (feat-dim)	**2.53**	**3.18**	**18.90**

Table 8: **Ablation experiments of the fusion method.** Best in **bold**.

shared decoder has larger errors but shorter runtime while the model using multi-FC decoders has smaller errors but suffers from poor real-time performance. The model using a Sparse-FC decoder has the advantages of both low error and fast running speed.

Fusion Method. The methods to integrate features from two pathways are worth exploring. We have explored three different methods for feature integration: addition, concatenation along the feature dimension, and concatenation along the joint dimension. As shown in Table 8, the fusion method of concatenating features along the embedding dimensions achieves the best performance.

Simulation Training. An important distinction of ReliaAvatar compared to other methods is its ability to simulate various scenarios during the training process. This is attributed to the model architecture with joint motion prediction and the autoregressive training paradigm. The results presented in Table 9 demonstrate that simulation training for data-loss scenarios does not adversely affect performance in the standard scenario.

4.6 Analysis

Ability of motion prediction. An important capability that we expect ReliaAvatar to achieve is the ability to predict full-body movements of the current frame using historical trajectory states when tracker signals are lost. This enables us to drive avatars smoothly even in the absence of real-time tracker information. We calculate MPJPE for the 40 frames after losing real-time tracker signals. The results presented in Table 10 demonstrate that ReliaAvatar is capable of predicting relatively accurate poses within several tens of frames after the disappearance of tracker signals.

method	Standard.	Instantaneous	Prolonged
no mask	**3.17**	51.41	24.30
+ random mask	3.18	3.92	12.26
+ prolonged mask	3.18	**3.51**	**5.17**

Table 9: Ablation experiments of simulation training. Best in **bold** and second best underlined.

Frame	#1	#3	#7	#10	#20	#30	#40
MPJPE	3.41	3.51	3.91	4.34	6.34	8.94	11.62

Table 10: The statistical errors after the interruption of tracker signals. #n represents the n-th frame after the interruption.

Inference Time. To ensure a fair comparison of the operational efficiency of different models, we conduct speed tests on state-of-the-art models under identical operating modes and hardware conditions. For each model, we construct a zero input and run it 10,000 times to obtain the average time as its inference time. All tests are conducted on a single GeForce RTX 3090 GPU. As shown in Figure 4, ReliaAvatar surpasses other models in both operational efficiency and accuracy.

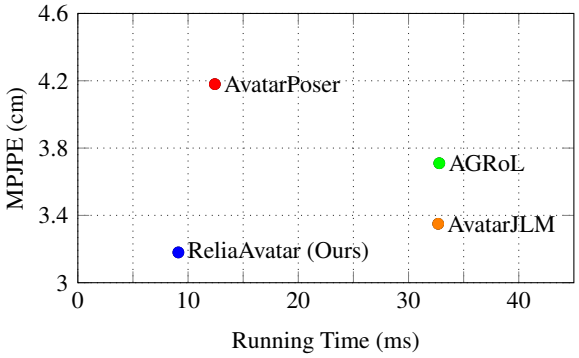


Figure 4: Inference time comparison.

5 Conclusion

We conducted a comprehensive exploration of the full-body avatar pose estimation problem under low-quality signal scenarios, which had not been systematically investigated before. We summarize three scenarios that may be encountered in practical applications: standard scenario, instantaneous data-loss scenario, and prolonged data-loss scenario. To address these challenges, we proposed *ReliaAvatar*, a real-time, robust, autoregressive avatar animator. We consider the full-body avatar pose estimation problem as a combination of joint upsampling and motion prediction. Therefore, ReliaAvatar possesses an upsampling pathway and a prediction pathway. Furthermore, we incorporate simulation training for data-loss scenarios on top of autoregressive training. Experimental results demonstrate that ReliaAvatar not only outperforms other methods in the data-loss scenarios but also achieves state-of-the-art performance in the standard scenario. In addition to its outstanding accuracy and robustness, ReliaAvatar also exhibits superior running efficiency compared to other methods. These advantages reduce the hardware requirements for animating a full-body avatar. As a result, more affordable and convenient devices can be used to drive avatars, which contributes to the wider adoption of related technologies.

Acknowledgements

This work was supported by National Key R&D Program of China under Grant No. 2021ZD0110400, the Fundamental Research Funds for the Central Universities No. xxj032023020, and sponsored by the CAAI-MindSpore Open Fund, developed on OpenI Community.

References

[Ahuja *et al.*, 2021] Karan Ahuja, Eyal Ofek, Mar Gonzalez-Franco, Christian Holz, and Andrew D Wilson. Coolmoves: User motion accentuation in virtual reality. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2):1–23, 2021.

[Aliakbarian *et al.*, 2023] Sadegh Aliakbarian, Fatemeh Saleh, David Collier, Pashmina Cameron, and Darren Cosker. Hmd-nemo: Online 3d avatar motion generation from sparse observations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9622–9631, 2023.

[Bai *et al.*, 2024] Yifan Bai, Zeyang Zhao, Yihong Gong, and Xing Wei. Artrackv2: Prompting autoregressive tracker where to look and how to describe. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[Di *et al.*, 2023] Xinhan Di, Xiaokun Dai, Xinkang Zhang, and Xinrong Chen. Dual attention poser: Dual path body tracking based on attention. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2795–2804. IEEE, 2023.

[Dittadi *et al.*, 2021] Andrea Dittadi, Sebastian Dziadzio, Darren Cosker, Ben Lundell, Thomas J Cashman, and Jamie Shotton. Full-body motion from a single head-mounted device: Generating smpl poses from partial observations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11687–11697, 2021.

[Du *et al.*, 2023] Yuming Du, Robin Kips, Albert Pumarola, Sebastian Starke, Ali Thabet, and Artsiom Sanakoyeu. Avatars grow legs: Generating smooth human motion from sparse tracking inputs with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 481–490, 2023.

[graphics lab., 2000] CMU graphics lab. Cmu graphics lab motion capture database. http://mocap.cs.cmu.edu/, 2000. Accessed: 2024-05-08.

[Huang *et al.*, 2018] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J Black, Otmar Hilliges, and Gerard Pons-Moll. Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018.

[Jain *et al.*, 2016] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 5308–5317, 2016.

[Jiang *et al.*, 2022] Jiaxi Jiang, Paul Streli, Huajian Qiu, Andreas Fender, Larissa Laich, Patrick Snape, and Christian Holz. Avatarposer: Articulated full-body pose tracking from sparse motion sensing. In *European Conference on Computer Vision*, pages 443–460. Springer, 2022.

[Jiang *et al.*, 2023] Jiaxi Jiang, Paul Streli, Manuel Meier, Andreas Fender, and Christian Holz. Egoposer: Robust real-time ego-body pose estimation in large scenes. *arXiv preprint arXiv:2308.06493*, 2023.

[Lehrmann *et al.*, 2014] Andreas M Lehrmann, Peter V Gehler, and Sebastian Nowozin. Efficient nonlinear markov models for human motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1314–1321, 2014.

[Liu *et al.*, 2019] Zhenguang Liu, Shuang Wu, Shuyuan Jin, Qi Liu, Shijian Lu, Roger Zimmermann, and Li Cheng. Towards natural and accurate future motion prediction of humans and animals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10004–10012, 2019.

[Ma *et al.*, 2022] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6437–6446, 2022.

[Mahmood *et al.*, 2019] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019.

[Mao *et al.*, 2019] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9489–9497, 2019.

[Mao *et al.*, 2020] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. History repeats itself: Human motion prediction via motion attention. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 474–489. Springer, 2020.

[Martinez *et al.*, 2017] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2891–2900, 2017.

[Müller *et al.*, 2007] Meinard Müller, Tido Röder, Michael Clausen, Bernhard Eberhardt, Björn Krüger, and Andreas Weber. Mocap database hdm05. *Institut für Informatik II, Universität Bonn*, 2(7), 2007.

[Shi *et al.*, 2022] Dahu Shi, Xing Wei, Liangqi Li, Ye Ren, and Wenming Tan. End-to-end multi-person pose estimation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11069–11078, June 2022.

[Taylor *et al.*, 2006] Graham W Taylor, Geoffrey E Hinton, and Sam Roweis. Modeling human motion using binary latent variables. *Advances in neural information processing systems*, 19, 2006.

[Troje, 2002] Nikolaus F Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of vision*, 2(5):2–2, 2002.

[Unity, 2018] Unity. The final inverse kinematics solution for unity. https://assetstore.unity.com/packages/tools/animation/final-ik-14290, 2018. Accessed: 2024-05-08.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Vlasic *et al.*, 2007] Daniel Vlasic, Rolf Adelsberger, Giovanni Vannucci, John Barnwell, Markus Gross, Wojciech Matusik, and Jovan Popović. Practical motion capture in everyday surroundings. *ACM transactions on graphics (TOG)*, 26(3):35–es, 2007.

[Von Marcard *et al.*, 2017] Timo Von Marcard, Bodo Rosenhahn, Michael J Black, and Gerard Pons-Moll. Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. In *Computer graphics forum*, volume 36, pages 349–360. Wiley Online Library, 2017.

[Wei *et al.*, 2023] Xing Wei, Yifan Bai, Yongchao Zheng, Dahu Shi, and Yihong Gong. Autoregressive visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9697–9706, 2023.

[Xsens, 2013] MVN Xsens. Full 6dof human motion tracking using miniature inertial sensors. *Daniel RoetenbergLuingeHenk*, 2013.

[Yan *et al.*, 2024] Kangwei Yan, Fei Wang, Bo Qian, Han Ding, Jinsong Han, and Xing Wei. Person-in-wifi 3d: End-to-end multi-person 3d pose estimation with wi-fi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024.

[Yang *et al.*, 2021] Dongseok Yang, Doyeon Kim, and Sung-Hee Lee. Lobstr: Real-time lower-body pose prediction from sparse upper-body tracking signals. In *Computer Graphics Forum*, volume 40, pages 265–275. Wiley Online Library, 2021.

[Yi *et al.*, 2021] Xinyu Yi, Yuxiao Zhou, and Feng Xu. Transpose: Real-time 3d human translation and pose estimation with six inertial sensors. *ACM Transactions on Graphics (TOG)*, 40(4):1–13, 2021.

[Zheng *et al.*, 2023] Xiaozheng Zheng, Zhuo Su, Chao Wen, Zhou Xue, and Xiaojie Jin. Realistic full-body tracking from sparse observations via joint-level modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14678–14688, 2023.

[Zhou *et al.*, 2019] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.