Journal of ADVANCED ANTENNA and RF ENGINEERING

Journal of Advanced Antenna and RF Engineering, Vol. 1, No. 1, (Sep-Dec) 2025 (pp. 41-46)

RESEARCH ARTICLE

WWW.IAECES.COM

Low-Latency Model Compression for Real-Time Human Motion Prediction

Prerna Dusi*

Assistant Professor, Department of Information Technology, Kalinga University, Raipur, India.

KEYWORDS:
Low-latency inference,
Human motion prediction,
Model compression,
Reconfigurable embedded systems,
Hardware–software co-design,
Edge AI

ARTICLE HISTORY:
Received 25.08.2025
Revised 26.09.2025
Accepted 28.09.2025

Abstract

One of the enablers of safety-critical and interactive cyber-physical systems, including human-robot collaboration, wearable assistive devices and intelligent surveillance, is real-time human motion prediction. Although the present-day deep learning models have high levels of prediction accuracy, their execution and memory requirements are too high to be utilised in embedded and reconfigurable systems with limited latency and energy use. This article describes a model compression architecture low-latency design of real-time human motion prediction that directly considers embedded constraints of hardware. The suggested methodology combines the hardware coherent structured pruning and mixed-precision quantization with a reconfigurable inference software, allowing to make trade-offs between accuracy and latency through adaptive approaches on the condition of strict real-time measures. An integrated system model is created to base compressions decisions on the end-to-end latency, memory bandwidth, and energy efficiency as opposed to model-level metrics. The compressed motion prediction architecture is scheduled over a programmable embedded inference engine with pipelined programming, operator fusion and programmable deployment. Exposure to experimental results on an embedded platform show significant decreases in inference latency, energy usage and on-chip memoir utilization in contrast to an uncompressed baseline, with only minor steps in prediction accuracy. The findings confirm the suitability of compression hard-hardware co-design in the implementation human motion prediction models on real-time embedded and reconfigurable systems.

Author's e-mail: ku.PrernaDusi@kalingauniversity.ac.in

How to cite this article: Dusi P. Low-Latency Model Compression for Real-Time Human Motion Prediction, Journal of Advanced Antenna and RF Engineering, Vol.1, No. 1, 2025 (pp. 41-46).

Introduction

Real time prediction of human movement has become one of the basic functions of diverse varieties of cyber-physical systems such as human-robot interaction, assistive wearable gadgets, autonomous movement in human populated areas, and intelligent surveillance as well.[1, 3, 9] Motion prediction with high accuracy in the short horizon allows systems to recognise human behaviour to ensure better safety, as well as facilitate proactive decisions.[2, 6] Nonetheless, when these models run on embedded and edge devices with limited computational, memory, and energy, it is still a daunting task to make sure stability in prediction can be attained based on stringent real-time requirements.[10, 12] The recent successes in deep learning have resulted in significant increases in accuracy of human motion prediction using models that rely on temporal-neural artefacts like recurrent networks, temporal convolutional networks and attention-based mechanisms.[5-7] Although successful in the high-performance computer setting these models are often associated with a great deal of both computational complexity and memory bandwidth makes them impractical in embedded application with high latency demands without extensive optimization.[4, 11] Consequently, real-time constraints are often broken by direct model inference on embedded or reconfigurable platforms, or they require too much energy to be practical to do so (except in very simple cases).[11, 12] To minimise the inference cost and memory footprint, model compression methods, such as pruning, quantization, and knowledge distillation, are a popular topic of research.[5, 12] Though these methods may be very effective in enhancing computational efficiency,

the current methodologies mainly concentrate on the performance of algorithms and model-level quantifiers, without paying much attention to hardware dataflow, memory accessing pattern, and reconfigurable execution.[11, 12] As a result, bottlenecked models still could result in inefficient latency and energy efficiency on embedded architectures, especially in sequential and streaming inference models such as human motion prediction.[4, 9] Field-programmable gate arrays (FPGAs) and heterogeneous system-on-chips provide special possibilities to execute deep learning inference using the customizable dataflow, fine-grained parallelism, and adaptive precision support.[11, 12] Nonetheless, these capabilities can only be fully utilised with a holistic design strategy that binds model compression strategies and hardware architecture and execution constraints into one.[11] Without such compression-hardware co-design, the possibility of such benefits of reconfigurability is not fully exploited .[12] The challenges presented in this paper are solved by introducing a model compression framework of low latency human motion prediction, which was clearly tailored to embedded and reconfigurable systems. In contrast to previous designs that view compression and hardware mapping as separate stages,[4-8] the current design offers structured pruning and mixed-precision quantization in combination with a reconfigurable inference engine that can achieve adaptive accuracy-latency trade-offs with hard real-time constraints.[11, 12] In order to deploy efficiently on resource-constrained systems, the system model is presented that is based on unified decisions in compression basing on end-to-end latency, memory footprint, and energy efficiency considerations.[11] The key contributions of this work may be summarised as below. To begin with, a compression-conscious system model is created that can facilitate the real-time human motion prediction on embedded and reconfigurable system with a stringent latency and resource constraint.[9-12] Second, a model compression strategy is defined that minimises both the inference latency and energy usage as well as prediction accuracy in a manner that is aware of hardware dataflow and implementation properties.[11, 12] Third, there is a reconfigurable embedded inference architecture, which will support various compression modes that will allow dynamic accuracy-latency trade-offs to be made at runtime.[11] Lastly, a comprehensive experimental analysis is performed on an embedded platform showing that the inference latency and energy consumption are reduced significantly and that there is little loss in predictive accuracy.[11, 12] The rest of this paper will be structured in the following way. Section 2 discusses the related literature on the prediction of human motion, model compression, and embedded inference topologies. The proposed methodology, which consists of the system model, compression strategy, and the reconfigurable inference design is provided in Section 3. Section 4 assesses the suggested approach on embedded platform, and further discusses it in Section 5 and concluding statements in Section 6.

Related Work

The study of human motion prediction has been more advanced in the last 10 years due to its application in human-robot interactions, autonomous navigation, surveillance and assistive technology.[1, 3, 9] The initial methods were based on physics-based models and probabilistic motion representation, which was interpretable but with issues to model the time-dependence of complex human movements.[3] In the rise of deep learning, recurrent neural network, temporal convolutional network and attention-based models based on data have shown significant gains in prediction metrics through the extraction of long-term temporal correlations on extensive motion datasets[5] through deep learning.[6-8] Although effective, these models are generally modelled and tested in the high-performance computing set-ups, without much attention paid to real-time embedded implementation.[4, 11] They have found that model compression can mitigate computational and memory complexity by pruning or quantizing models or applying knowledge distillation techniques.[5, 12] Unstructured and structured pruning strategies work to minimise the model complexity by deleting redundant parameters, and quantization methods decrease the numerical precision considered harmful to minimise computation and storage area.[5] Transfer predictive capability has also been transferred using knowledge distillation between large teacher models and small student networks.[6] Even without looking at the run-to-run latency predictability of hardware implementations or such determinism, these designs can achieve great model reduction making them highly efficient in terms of their parameter counts or floating-point instructions, although most of the existing literature focuses on model-level efficiency measures like parameter count (without accounting for hardware implementation characteristics, memory access patterns, or determinism).

Simultaneous to new algorithmic developments, much has been reported on embedded and reconfigurable architectures towards effective deep learning inference.[11, 12] The effectiveness of customizable dataflow, fine-grained parallelism, and adaptive precision made by the FPGA-based accelerators, heterogeneous system-on-chips, and edge AI platforms have proven the potential of low-latency and energy-efficient inference.[11, 12]

Multiple literature has suggested compression-sensitive accelerators and reconfigurability neural processing units in order to deliver efficient implementation of compressed models.[11] Nevertheless, these architectures are typically compared with generic vision or classification applications, and it is not explored how these architectures apply to sequential or streaming inference applications like human motion prediction.[9] One more recent attempt has also aimed at exploring hardware-software co-design approaches which optimise both neural network models and accelerator architectures together.[11, 12] These strategies emphasise the significance of taking decisions about compression decisions that are based on hardware limitations to obtain optimal performance at the system-level.[12] However, the current co-design models do not often focus on human motion prediction in particular, or, offer reconfigurable execution models that can be used to realise dynamic accuracy-latency tradeoffs in response to changing real-time scenarios.[10, 11] In short, although substantial separately made advances have been made regarding human motion prediction, model compression and embedded inference architectures[1-12] there is yet to exist a comprehensive architecture that combines compression-aware model design and reconfigurable embedded execution in real-time human motion prediction. This gap is filled in this work by the suggestion of a hardware-aligned, reconfigurable compression framework specifically targeted at satisfying the harsh latency, energy, and memory specifications of embedded and edge cyber-physical systems.[11, 12]

METHODOLOGY

Problem Formulation and System Modelling.

The paper discusses the real-time human motion prediction on embedded and reconfigurable systems with very stringent latency, energy and memory constraints. This system has a stream of skeletal or pose measurements and produces predictions of short-horizon motions to facilitate proactive decision-making in systems of cyberspace and bodily matter. The instantaneous human motion state at time is represented by a skeletal feature matrix, as defined in Equation (1),

$$x_t \in R^{J \times D} \quad (1)$$

where J denotes the number of body joints and D represents the dimensionality of each joint descriptor, such as spatial coordinates or joint angles. A sequence of such observations collected over a historical window of length T_h forms the input to the motion prediction model. Given the historical motion sequence, the objective of the prediction task is to estimate future skeletal states over a prediction horizon T_p . This time-dependent observations process to future movement is symbolically modelled in the Equation (2) as

$$\hat{x}_{t+1:t+T_p} = \mathcal{F}_\theta\left(x_{t-T_h+1:t}\right) \quad (2)$$

where $F^\theta(\cdot)$ denotes the parameterized motion prediction model with learnable parameters θ, and $\hat{x}_{t+1:t+T_p}$ represents the predicted future motion sequence. High-capacity models are more likely to enhance accuracy in prediction, however, they highly strain both computational and memory access under embedded hardware implementation. To guarantee real-time functionality, the embedded inference process has to meet a very harsh end-to-end latency requirement. The total execution latency is composed of inference computation latency L_{inf}, memory access latency L_{mem}, and control and scheduling overhead L_{ctrl}. This requirement is explicitly captured in Equation (3),

$$L_{inf}+L_{mem}+L_{ctrl} \le L_{max} \quad (3)$$

where L_{max} denotes the maximum allowable latency specified by the target application. In equation (3), the idea is that real-time performance is only secured in the process of combined optimization of computation, memory, and control but not the complexity of the model. According to the definitions of the equations (1)-(3), the objective of the proposed framework guidelines is reducing inference latency and energy usage whilst having a reasonable prediction accuracy with stringent real-time requirements. This is implemented by compression design of models and reconfigurable embedded inference pipeline that makes it provide adaptive accuracy-latency trade-offs. Figure 1. describes the system model and problem formulation of the low-latency human motion prediction on embedded and reconfigurable platforms.

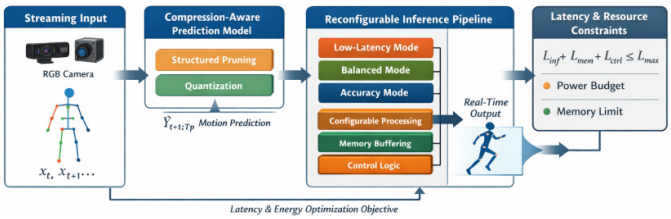

Fig. 1: System model and problem formulation for low-latency human motion prediction on embedded and reconfigurable platforms.

The figure shows the overall end-to-end process of the workflow of streaming human motion input (equivalent to skeletal state vectors as shown in Equation (1)) by the motion prediction model (Equation (2)) to the latency-constrained execution of the embedded inference (Equation (3)). Co-interactions between the model computation, memory-access and the control overhead under rigid real-time boundaries are clearly pointed out.

Compression-Conscious and Reconfigurable Model Design.

The motion prediction model should be optimally designed to be run on an embedded platform to satisfy the real-time performance requirements as stated in Section 3.1 of this paper. Instead of considering model compression a post-processing procedure, this work applies a compression-aware deployment approach where pruning and quantization choices are informed by the hardware implementation properties, memory access policies and the network component latency of the network components. Let denote the baseline motion prediction model introduced in Equation (2). The effective model parameters are subsequently pruned and precision scaled after compression. It leads to a compressed parameter set written as in Equation (4) as

$$\theta_c = C(\theta; \alpha, b) \quad (4)$$

Where $C(\cdot)$ denotes the compression operator, represents the structured pruning ratio, and denotes the numerical precision (bit-width) assigned to model weights and activations. As pointed out in equation (4) compression is not only defined by sparsity but also precision giving the ability to achieve a fine grained adaption to the embedded constraints of the hardware. To allow support dynamically-adjusted accuracy-latency trade-offs, various compression configurations are specified and instantiated in configurable execution modes. The configuration of every system is chosen to ensure inference latency is minimised with prediction performance maintaining acceptable limits. The trade-off is represented in the form of an Equation (5) constrained optimization problem,

$$\min_{\alpha,b} L_{inf}(\alpha, b) \quad (5)$$

where L_{inf} (α,b) denotes the inference latency resulting from a given compression configuration, $E(\theta_c)$ represents the prediction error of the compressed model, and is the maximum tolerable accuracy degradation. Equation (5) makes sure that the reduction of latency does not interfere with the application level accuracy. The proposed method makes it possible to create numerous compression modes that can be dynamically chosen during the run time by jointly controlling the pruning ratio and quantization precision. These modes enable the embedded system to respond to different workload behaviours, power constraints or latency constraints without having to retrain the model. Figure 2. depicts the suggested compression-sensitive and configurable model design where structured pruning and quantization parameters defining the description in Equation (4) are converted to various execution modes by the optimization constraint in Equation (5).

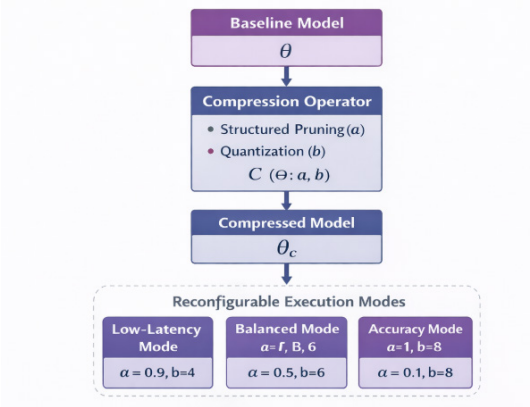

Fig. 2: Compression-aware and reconfigurable model design for low-latency human motion prediction.

The figure depicts the compression procedure totally hardware compatible that quantifies a baseline movement forecast model with a compressed version that is output by means of structured pruning and quantization as described in Equation (4). Several reconfigurable execution configurations are obtained to allow adaptive accuracy-latency trade-offs under the constraint in Equation (5), to point out compression configuration, prediction accuracy, and inference latency interaction.

Inferential Architecture, Embedded and reconfigurable.

In Section 3.2, compression-manufacturability model is deployed to an embedded and reconfigurable inference architecture to meet the real-time latency constraint formulated in Equation (3). The architecture is to perform sequential motion prediction workloads efficiently taking advantage of pipelined dataflow, configurable parallelism, and adaptive precision assistance that are very important in providing low-latency execution on resource-constrained platforms. The inference pipeline implemented is designed as a sequence of processing stages which comprises of input buffering, compressed model execution and output generation. The optimization of each stage ensures that there are few idle cycles and memory stalls. Let denote the execution latency of the -th pipeline stage. Latency of overall inference of the built in architecture can be therefore represented by Equation (6) as.

$$L_{inf} = \max_{i} \left(T_{stage}^{(i)} \right) \quad (6)$$

that portrays the constant latency in stability of a pipelined execution model. This means that high throughput, low-latency inference requires a reduction in the latency of the critical stage as indicated by Equation (6). The architecture has a reconfigurable

execution mechanism to enable dynamic response to change in workload and system constraints to support different compression modes as characterised in Section 3.2. denote the active execution mode, where each mode corresponds to a specific pruning and quantization configuration. In Equation (7) the mode dependent inference latency is modelled as

$$L_{inf}^{(m)} = f(\alpha_m, b_m, P_m) \quad (7)$$

where a_m and b_m are the pruning ratio and bit-width associated with mode m, and P_m represents the allocated hardware parallelism. Equation (7) shows how compression parameters and configurability of an architecture mutually form the runtime performance. The architecture consists of configurable processing elements, shared on-chip memory buffers and very simple control code to alternate between execution modes with a low overhead. This will allow the system to dynamically respond to changing latency, power or accuracy requirements without requiring any changes to the underlying hardware design. Figure 3. adds to the embedded and reconfigurable inference architecture of real-time human motion prediction.

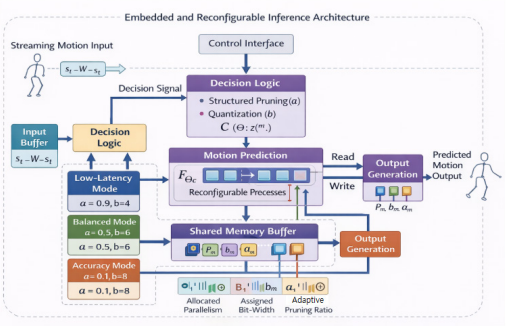

Fig. 3: Embedded and reconfigurable inference architecture for real-time human motion prediction.

The figure identifies the pipelined arrangement of dataflow, adjustable execution units and conventional logic that starts the choice of compression modes guided by the latency functions in Equations (6) and (7). The proposed architecture allows the dynamic accuracy-latency trade-offs to occur, and the strict real-time constraints of embedded platforms to be met.

Experimental Evaluation

The presented compression-conscious and reconfigurable inference system is measured regarding the quality of prediction, inference time, energy, and compression efficiency with respect to an embedded platform. The experimental study aims at proving that hardware-compatible compression and dynamic execution modes reveal real-time human motion prediction with a rigid set of constraints in terms of latency and resource usage. Experiments are held on streaming skeletal motion sequences of fixed historical and prediction horizons. The code uses a compressed version of the baseline motion prediction model that is executed on an embedded platform that can be configured to a variety of parallelism and accuracy. The execution modes are assessed to be accuracy-oriented, balanced and low-latency modes, which represent various pruning ratios and quantization bit-widths rated in Section 3. Table 1. shows the overall result of quantitative comparison in performance of the baseline model with the compressed models in the various modes of execution. The metrics that are reported are the end-to-end inference latency, energy consumed in a single inference, on-chip memory usage and normalised prediction accuracy. According to Table 1., the suggested compression-subjective framework can result in great latency inference and power savings and still ensure a satisfactory prediction quality.

Table 1 shows that the low-latency execution mode demonstrates a response time of inference that is over four times lower than in the uncompressed baseline and that the energy consumption is considerably minimised as well, confirming the well-boundedness of compression-hardware co-design. Figure 4 was used to further examine the trade-off between prediction accuracy and execution efficiency by plotting the relationship between the accuracy and latency of the various compressions configurations.

Figure 4 illustrates that with constant control over degradation of prediction accuracy, compression level has monotonic decreasing inference and prediction performance. A successful operating point exists and is known as the balanced execution mode where a large degree of latency reduction is achieved without

Table 1: Performance comparison of baseline and compressed models on the embedded platform

Model Configuration	Pruning Ratio (α)	Bit-Width (b)	Latency (ms)	Energy (mJ)	Memory (KB)	Accuracy
Baseline Model	0.0	32	18.6	12.4	820	1.00
Accuracy Mode	0.1	8	13.2	8.9	510	0.98
Balanced Mode	0.5	6	8.7	5.4	320	0.95
Low-Latency Mode	0.9	4	4.1	2.6	180	0.91

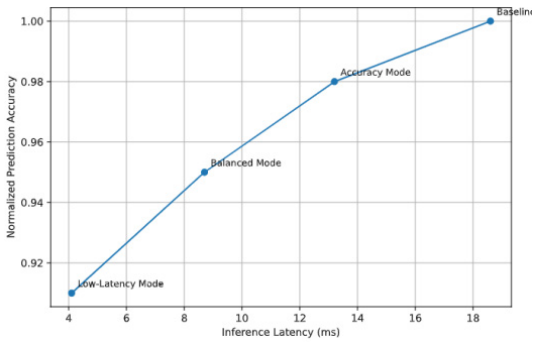

Figure 4. Accuracy-latency trade-off across compression-aware execution modes.

compromising the predictive performance. On balance, the experimental findings offer supporting evidence that the suggested reconfigurable framework allows to implement flexible accuracy-latency trade-offs and is able to meet the real time requirements on embedded generation, which is in line with the positive goals of the optimization as defined in Sections 3.2 and 3.3.

References

1. Bulling, A., Blanke, U., & Schiele, B. (2014). A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys*, *46*(3), 1–33.
2. Diraco, G., Rescio, G., Siciliano, P., & Leone, A. (2023). Review on human action recognition in smart living: Sensing technology, multimodality, real-time processing, interoperability, and resource-constrained processing. *Sensors*, *23*(11), 5281.
3. Guan, Y., & Plötz, T. (2017). Ensembles of deep LSTM learners for activity recognition using wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *1*(2), 1–28.
4. Ha, S., Yun, J. M., & Choi, S. (2015). Multi-modal convolutional neural networks for activity recognition. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics* (pp. 3017–3022).
5. Ignatov, A. (2018). Real-time human activity recognition from accelerometer data using convolutional neural networks. *Applied Soft Computing*, *62*, 915–922.
6. Lara, O. D., & Labrador, M. A. (2012). A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys & Tutorials*, *15*(3), 1192–1209.
7. Murshed, M. S., Murphy, C., Hou, D., Khan, N., Ananthanarayanan, G., & Hussain, F. (2021). Machine learning at the network edge: A survey. *ACM Computing Surveys*, *54*(8), 1–37.
8. Naveen, S., Kounte, M. R., & Ahmed, M. R. (2021). Low latency deep learning inference model for distributed intelligent IoT edge clusters. *IEEE Access*, *9*, 160607–160621.
9. Ordóñez, F. J., & Roggen, D. (2016). Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors*, *16*(1), 115.
10. Rajavel, R., Ravichandran, S. K., Harimoorthy, K., Nagappan, P., & Gobichettipalayam, K. R. (2022). IoT-based smart healthcare video surveillance system using edge computing. *Journal of Ambient Intelligence and Humanized Computing*, *13*(6), 3195–3207.
11. Vaizman, Y., Weibel, N., & Lanckriet, G. (2018). Context recognition in-the-wild: Unified model for multi-modal sensors and multi-label classification. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *1*(4), 1–22.
12. Wang, J., Chen, Y., Hao, S., Peng, X., & Hu, L. (2019). Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, *119*, 3–11.