UNIVERSITY OF PADUA

DEPARTMENT OF ENGINEERING AND INDUSTRIAL SYSTEMS MANAGEMENT
MASTER'S DEGREE IN MECHATRONIC ENGINEERING

MASTER'S THESIS

SEMANTIC GUIDED MULTI-FUTURE HUMAN MOTION PREDICTION

Supervisor: Stefano Michieletto

Co-supervisor: Michael Vanuzzo

Co-supervisor: Mattia Guidolin

Student: Francesco Borsatti
2007675-IMC

ACADEMIC YEAR: 2022-23

ABSTRACT

The primary objective of this thesis is to enhance the accuracy of a machine learning model for predicting multiple potential future human movements. This has been achieved by incorporating semantic information into the input data, contributing to the broader goal of advancing safer and efficient human-robot cooperation within an Industry 5.0 context.

We analyzed diverse performance metrics and explored multiple aspects of the problem, from the integration of semantic data into the preprocessing phase, to the development of semantic class labeling strategies and comprehensive evaluation methodologies.

We demonstrated the significance of semantic context in motion prediction through a comparative analysis of models utilizing kinematic data alone and those augmented with semantic information. We conducted experiments on one of the most recent and significant datasets in the literature, simulating a realistic scenario where approximately 40% of the data lacked semantic information. Remarkably, even in this setting, the model with the most parameter capacity enhanced by semantic information (*128 kin+sem*) outperformed both the kinematic-only counterpart and the zero-velocity baseline model.

In particular, "*128 kin+sem*" reduced by 3% the cumulative error over the "*128 kin*" and exhibited a 10% error reduction against the zero-velocity over one second of prediction timespan.

The importance of a careful model design is highlighted, by showing why ensuring sufficient parameter capacity is necessary to effectively accommodate the augmented input data when semantic information is introduced.

Regarding the practical applications of our model, it is important to consider that for cooperative robotic planning, the initial moments of motion prediction hold relatively less significance. The primary focus lies in achieving accurate predictions for a range of potential outcomes in long-term motion prediction.

While our research has primarily centered around cooperative robotic applications, we also expect that our methodologies can be applied to diverse fields beyond the initial scope. The prediction of future body movements opens up possibilities for offline utilization in non-real-time tasks, such as generating realistic human motion. In particular, motion prediction models hold potential in generating partial movements, allowing us to leverage a limited amount of available data to generate new data points.

In terms of performance evaluation, we measured the time it took to compute the inference of predictions using the metrics script. The tested models exhibited slightly longer inference times compared to the duration of the predicted sequence, since the models were designed with offline testing in mind, but can be optimized for real-time with software and hardware adaptations.

The findings of this study lay the foundation for future research endeavors, as numerous deep learning models that solely rely on kinematic information could potentially achieve groundbreaking results by effectively incorporating semantic information. This study represents an initial step in showcasing the influential role of semantics in enhancing the prediction of human motion.

The problem with quotes
found on the internet is that
they are often not true.
— Abraham Lincoln

ACKNOWLEDGMENTS

I would like to extend my heartfelt gratitude to the individuals who have played a significant role in my thesis journey. Their support, guidance, and contributions have been invaluable, and I am truly grateful.

First, I would like to express my deepest appreciation to Stefano Michieletto for providing me with this research endeavor, and I am truly grateful for his expertise and support.

I would also like to thank Monica Reggiani for her advice throughout the process.

A special mention goes to Mattia Guidolin and Michael Vanuzzo, who have been my senpais, guiding me and sharing their knowledge and experience.

I need to mention my dear colleagues Davide Carollo for optimizing parts of the code, making it up to 10x faster; Marco Casarin for his contributions in fixing the tail problem in the mocap visualizer and help with the zero velocity metrics; and Matteo Cunico for providing me with the problem on which to work.

A heartfelt thanks to all my friends and family who have supported me throughout this journey.

Lastly, I would like to express my deepest appreciation to my love, Francesca, for always being there for me. You are my constant source of joy and motivation.

To each person mentioned here, and to those who have supported me in countless other ways, I extend my sincerest thanks.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 PROBLEM DESCRIPTION

1.1.1 *Collaborative robotics*

The industrial sector is currently experiencing a revolution known as Industry 5.0, where the focus is on environmental and social factors, with an increased emphasis on the cooperation between humans and robots. The idea behind this shift is to create an environment where humans and robots can coexist and leverage each other's strengths.

An extensive review on quality in Human-Robot Interaction (HRI) applications in manufacturing environments [11] discusses the different interests in the robotics community, such as performance-centered and human-centered paradigms, emphasizing the need to enhance human work and well-being in robotics systems. This review asserts that robots can often handle repetitive, unsafe, and physically demanding tasks, while humans engage in critical thinking and customization, highlighting the strengths of human-robot interaction.

Human-robot cooperation in an industrial environment can be a significant advantage. However, it is important to consider various factors that may impact the efficiency and effectiveness of this cooperation. Here are some key points to consider:

- *Efficiency*: In some cases, human-robot cooperation may be slower than using a regular human worker. This could be due to several reasons, such as the complexity of the task, the need for coordination between humans and robots, or the limitations of current robotic technology.

- *Regulations*: Strict regulations surrounding the coexistence of humans and robots in the same working space can impact the efficiency of human-robot cooperation. These regulations are in place to ensure the safety of human workers and prevent accidents. However, they can also impose constraints on the movement and capabilities of robots, which may affect their performance [36]. Standards such as EN ISO 10218 aim to give guidelines for the implementation of hybrid production systems, where robots and humans can work together safely.

- *Possible improvements*: The performance of human-robot cooperation can be improved by making robots smarter and enabling them to predict the motion of human agents [25]. This requires advanced sensing, perception, and prediction capabilities in robots. By understanding human behavior and intent, robots can anticipate and adapt to human actions, leading to better coordination and efficiency.

1.1.2 *Anticipation of human motion*

To enable an efficient and safe human-robot cooperation, systems should accurately predict human actions and understand non-verbal cues. Neural network-based approaches have shown promise in predicting human actions, such as explored in an article by P. Schydlo et al. which research validates the use of gaze and body pose cues for action prediction in the context of human-robot cooperation [42]. Their proposed solution employs an encoder-decoder recurrent neural network model, which in part is similar to the architectures used in this thesis main model.

When cooperative robotic systems can anticipate human motion, they have a higher chance of avoiding collisions, ensuring a safer working environment. By proactively responding to human actions, robots can better coordinate and synchronize their actions with their human counterparts, reducing the time spent waiting for the other agent to complete their part of the task.

Multi-Future prediction

Human motion is a stochastic sequential process with a high level of intrinsic uncertainty. Given an observed sequence of poses, a diverse set of future pose sequences is likely to occur. Hence, due to the intrinsic uncertainty, even with an excellent model, when predicting a long-term sequence of future poses, it is improbable that the predictions for distant future poses will precisely match the ground truth as stated by Pavllo et al. [35].

For that reason, this work utilizes a machine learning model capable of generating multiple possible outputs from a single input. This approach is expected to enhance safety in the context of human-robot collaboration, as it enables proactive measures to mitigate potential risks.

1.2 METHODS

1.2.1 *Semantics in the context of human movement*

The integration of semantics into motion prediction models holds potential in various domains, particularly in collaborative robotics. By augmenting the prediction process with contextual meaning, these models can enhance the overall performance and adaptability of robotic systems operating in dynamic environments alongside humans. This can enable more efficient human-robot interaction, leading to improved collaboration, safety, and productivity.

The source of semantic information

As mentioned earlier, this thesis focuses on exploring the incorporation of semantic information to improve the accuracy of motion predictions. However, it is based on the fundamental assumption that this semantic data is accessible and readily available during the inference phase.

This assumption can be supported by the potential integration of an auxiliary model that identifies the semantics of actions through action recognition techniques. Action recognition in video footage using machine learning has been the subject of extensive research. Notably, K. Simonyan et al. demonstrated the effectiveness of deep neural networks for video classification in their work titled "Two-Stream Convolutional Networks for Action Recognition in Videos" [43]. Their study validated the usage of deep neural networks by achieving state-of-the-art performance on standard benchmarks at that time.

In more recent research, S. Yan, Y. Xiong et al. introduced a novel approach to modeling human-skeleton dynamics in their work titled "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition" [48]. They employed techniques such as pose estimation and action classification to effectively capture the dynamics of human-skeleton movements. This approach showcased promising advancements in the field of action recognition.

Another justification for assuming the availability of semantic information for the motion prediction, is that the context of human movement in a work environment can be derived from the scheduling of tasks performed by the operator in their activity plan. In such scenarios, the actions to be executed are often known and repetitive, offering a unique advantage for motion prediction. By incorporating knowledge of the expected actions, the prediction model can not only consider the spatio-temporal aspects of the body's movement trajectories in space, but also the high-level context associated with the agent's actions.

This higher-level comprehension of the action's significance provides the prediction model with valuable insights into the underlying intentions and goals of the operator. Consequently, it allows the model to generate more accurate and contextually relevant predictions of the agent's future movements. By considering not only the physical aspects of the motion but also the semantic context in which the actions occur, the prediction model gains a more comprehensive understanding of the human movement dynamics.

1.2.2 *Machine Learning, Artificial Intelligence*

To achieve the objective of building a prediction model for human motion, machine learning and artificial intelligence methodologies are used, leveraging their capabilities to process and interpret complex spatio-temporal data along with contextual information.

Why machine learning

Human motion is a really complex phenomenon, with sophisticated behavior and contextual dependencies. A mathematical model that accurately represents such phenomenon would be equally complex, even more so if the input data is strongly decimated in comparison to the real world.

This leads to the consideration that defining a model that predicts human motion would be a highly complex task. If a human had to come up with all the rules and exceptions that model the motion, it would take a very long time and incredible effort.

Since creating a model "manually" is not realistically achievable, machine learning (ML) can be a good choice for creating a model for a phenomenon that is very complex to model, in comparison to traditional methods of modeling physical systems.

ML has the capability to learn from large datasets and automatically discover meaningful relationships, enabling it to effectively model the complex and nonlinear nature of human motion and the relative semantic data.

While machine learning holds promise, it is important to note that there is no one-size-fits-all solution. ML includes a diverse range of architectures, each suited for different types of tasks and data.

Machine learning is not a modern concept, earlier architectures such as decision trees, support vector machines, and linear regression were employed before the advancements in computational power that facilitated the emergence of deep learning algorithms. These early (non-deep) machine learning approaches laid the foundation for understanding patterns and making predictions based on data. An example of three non-deep learning machine learning algorithms are listed below.

- *Decision Trees*: hierarchical structures that partition the input space based on features to make predictions or decisions. They are commonly used for classification and regression tasks in various domains.

- *Support Vector Machines* (SVM): supervised learning algorithm that finds an optimal hyperplane to separate different classes in the input space. SVMs are frequently applied to classification problems, especially when dealing with complex and high-dimensional data, but can also be used for regression tasks.

- *Linear Regression*: also a supervised learning algorithm and predicts a continuous target variable based on input features, assuming a linear relationship between them. Linear regression is often used for tasks such as predicting housing prices or estimating sales.

Deep learning architectures span from simpler ones like fully connected layers to more complex ones such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), transformers and Generative Adversarial Networks (GANs), to mention just a few examples. The choice of architecture depends on various factors such as the nature of the problem, the characteristics of the data, and the specific objectives of the task at hand. Thus, careful consideration and experimentation are essential in selecting the most appropriate ML architecture for a given scenario.

For example, image object recognition tasks are often effectively handled by Convolutional Neural Networks (CNNs). They consist of convolutional layers that can efficiently extract features from images by applying filters across the input. This capability allows CNNs to capture spatial hierarchies and patterns present in images, enabling them to discern complex visual features such as edges, textures, and shapes.

On the other hand, Recurrent Neural Networks (RNNs) have proven to be beneficial for speech recognition tasks. RNNs are designed to handle sequential data, such as audio waveforms, by processing information in a sequential and temporal manner. This sequential processing enables RNNs to capture dependencies and patterns over time, making them suitable for tasks that involve temporal sequences.

Inference times and physical feasibility

This study aims to investigate whether the semantic context of human agent actions can assist in prediction, without considering the physical feasibility of real-time implementation or other computational power-related issues, as it is a theoretical work. In this context, the term "real-time" refers to the requirement of performing inference for the human motion prediction, within a narrow time frame, as any delay beyond the specified timeframe would make the prediction irrelevant in practice.

Nevertheless, it is possible for machine learning models to operate in real-time if adequately optimized and paired with suitable hardware, as demonstrated in other application fields such as real-time object detection [38], licence-plate recognition [3], just to mention a few examples.

1.3 STATE OF THE ART

The problem at hand is: semantic multi-future human motion prediction. There have been numerous works on human body pose prediction, many of them focused on the kinematic aspect of the problem while not considering the semantic information, these approaches are discussed in the Section 1.3.1.

Those works still contain very insightful observations on the best ways on how to leverage kinematic information (usually body joint angles or body keypoint position) to predict motion. In this thesis work, the objective is to enhance the predictive ability of a model with the addition of semantic information, this is why considering existing approaches, although they do not employ semantics, is a good starting point.

SCAFF

One recent work that used both semantic and kinematic information is the paper titled "Semantic Correlation Attention-Based Multiorder Multiscale Feature Fusion Network for Human Motion Prediction" [23].

While this study, proposing a new approach named SCAFF for human motion prediction, does not explicitly account for the semantics as we intend in this work which is the type of action being performed (e.g., walking, playing sports), it does consider the semantic correlations in kinematic relations between body parts such as joints and bones.

Although those are different kinds of semantics, this is still noteworthy as it signifies a growing interest in the scientific community towards incorporating semantic information to enhance human motion prediction.

In SCAFF, the approach is to build a model able to assess the complex differences in joint and bone movements, with a set of operators that are deployed to extract patterns from this data. A semantic correlation attention module refines these patterns, focusing on the temporal relationships between different body parts.

The refined data is then fused to generate a comprehensive representation of the individual's current motion, serving as the basis for future motion prediction. Experimental results indicate that this method outperforms existing models, underscoring the potential value of incorporating semantics into motion prediction frameworks.

1.3.1 *Review of existing methods for human motion prediction*

QuaterNet

One of the most cited papers is titled "QuaterNet: A Quaternion-based Recurrent Model for Human Motion" [35]. It presents a new approach to improve the prediction and generation of 3D human pose sequences. Pavllo et al. found that joint rotation predictions can suffer from error accumulation along the kinematic chain and discontinuities in Euler angle or exponential map parameterization. Joint position methods require re-projection onto skeleton constraints to avoid issues like bone stretching and invalid configurations.

To address these limitations, the authors propose QuaterNet, a recurrent neural network that uses quaternions for rotation representation and a new loss function. The loss function conducts forward kinematics on a parameterized skeleton, which allows penalizing absolute position errors instead of angle errors. This combination leverages the advantages of joint orientation prediction with a position-based loss.

The experimental results demonstrate that for short-term predictions, QuaterNet quantitatively improves the (at the time) state-of-the-art by reducing angle prediction errors on the Human3.6m benchmark.

ST-Transformer

The paper "A Spatio-temporal Transformer for 3D Human Motion Prediction" [2] introduces a Transformer-based architecture for the task of generative modeling of 3D human motion.

In the past, recurrent neural networks (RNNs) and convolutional neural networks (CNNs) have been commonly used given their capacity to handle temporal data. However, these methods often neglect structural priors, and error accumulation over time can lead to non-plausible pose predictions.

The authors of this paper propose a Spatio-temporal Transformer (ST-Transformer) model, which learns a spatio-temporal representation explicitly without relying on a hidden state or fixed temporal encodings. The novel aspect of this model lies in the decoupling of the temporal and spatial dimensions, which allows for a spatio-temporal attention mechanism, thus capturing dependencies explicitly, mitigating error accumulation over time, and increasing interpretability.

Experimental results demonstrate that the ST-Transformer can outperform state-of-the-art models in short-term horizons and produce convincing long-term predictions (up to 20 seconds for periodic motions), such as locomotion. It demonstrates effectiveness in learning representations for both short-term and long-term motion predictions, which makes it a promising approach for 3D human motion prediction tasks.

MotionFlow

In the paper titled "Flow-based Spatio-Temporal Structured Prediction of Dynamics" [49], the authors introduce an end-to-end deep learning architecture that's tailored to learn temporal and spatial dependencies, enabling explicit modeling of joint connectivity. They also highlight that their model successfully marries the advantages of both stochastic and deterministic representations, enabling more reliable long-term predictions.

The methodology is designed to cope with the large variations of potential outputs, without losing dynamic diversity across body joint components. The presented MotionFlow model is a conditional autoregressive flow-based solution designed to learn spatio-temporal relations in dynamic systems. It excels at directly modeling the log-likelihood of temporal and spatial information for long sequences, yielding more robust spatio-temporal representations while preserving the structure of high-dimensional data.

Diverse and controllable motion prediction

In the publication titled "Generating Smooth Pose Sequences for Diverse Human Motion Prediction" [31], the authors introduce a deep generative network designed for both diverse and controllable motion prediction. They exploit the notion that human motions are fundamentally sequences of smooth, valid poses, and create a generator that predicts motion for different body parts sequentially. The model incorporates a normalizing flow-based pose prior and a joint angle loss to ensure the realism of the generated motion.

They offer a method for diverse motion prediction that doesn't rely on learning several mappings, providing an end-to-end trainable solution. This approach results in a fully controllable motion prediction, permitting the motion of certain portions of the human body to be fixed while generating diverse predictions for the remaining portions. The technique involves the application of a pose prior and a strict constraint on the predicted poses to form smooth sequences that satisfy human kinematic constraints, rather than learning a motion prior, which often suffers from a lack of sufficiently diverse training data. Notably, the pose prior is modeled as a normalizing flow, which enables exact computation of the data log-likelihood and further promotes diversity by maximizing the distance between pairs of samples during training.

Through experimental validation on two standard benchmark datasets (Human3.6M and HumanEva-I), the authors demonstrate that their approach outperforms state-of-the-art baselines in terms of both sample diversity and accuracy. Overall, the authors' contributions are two-fold: they present a unified framework achieving both diverse and part-based controllable human motion prediction, and they propose a pose prior and a joint angle constraint that regulate the training of the generator, encouraging it to produce smooth pose sequences.

Coordinated Motion Optimization

The paper "Prediction of Human Full-Body Movements with Motion Optimization and Recurrent Neural Networks" [21] presents a new approach for predicting complex human behaviors, particularly when those behaviors change in different environments. The proposed framework uses a dual strategy, utilizing a recurrent neural network to encode short-term body dynamics, while considering environmental constraints via gradient-based trajectory optimization.

The key premise of the work is to improve safety and efficiency in human-robot interaction by accurately predicting human motion. One limitation in other approaches is that the motion prediction model is trained only on human body motion data, and doesn't adequately consider scene context such as targets or obstacles.

The approach in this paper uses a recurrent neural network for pure kinematic predictions of human motion, but also adds to the model a way to control human velocities at each prediction step. The motion is then optimized using a gradient-based optimization algorithm.

One of the benefits of this method is that it's flexible, allowing for the integration of numerous constraints in motion planning, like smoothness, obstacle avoidance, and hand orientation.

Through experiments conducted on real motion data, the authors demonstrate that their framework significantly improves prediction accuracy compared to conventional neural network predictors, especially in long-term prediction.

1.3.2 *Human motion datasets*

In the field of body motion datasets, several alternatives have been developed, each with its unique characteristics and applications. The following sections contain a concise overview of these alternatives, outlining their key attributes, advantages, and potential limitations.

AMASS

Archive of Motion Capture As Surface Shapes [28] is a large and diverse collection of human motion data, aggregating 15 different optical marker-based mocap datasets into a common framework and parameterization. The major advantage is its size, featuring more than 40 hours of motion data, over 300 subjects, and more than 11,000 motions. This provides a vast resource for deep learning applications.

The dataset is standardized using the Skinned Multi-Person Linear Model (SMPL) [27], ensuring consistency and compatibility across various motions.

This dataset is particularly interesting for the purposes of this work, as in combination with the BABEL (Bodies, Action and Behavior with English Labels) [37] dataset, it addresses the challenge of understanding the semantics of human movement by providing precise descriptions for about 43 hours of mocap sequences from AMASS.

DIP

Deep Inertial Poser [19] is a deep learning model that can predict full-body pose from sparse (only six) Inertial-Measurement-Units (*IMUs*) in real-time. It has been trained on a large scale synthetic dataset generated from other datasets MoCap data (from AMASS), and fine-tuned on DIP-IMU, a real IMU dataset. DIP is especially useful for applications like AR and VR due to its real-time capabilities. This dataset has also been used in the project reported in the ST-Transformer paper.

H3.6M

Human 3.6 Million [20, 8] is one of the first and most well-established datasets in the field, containing 3.6 million 3D human poses and corresponding images. The data features 11 professional actors performing in 17 different scenarios. It is a rich dataset that is highly diverse and offers a good balance of male and female performers. One drawback is that larger datasets like AMASS are available, and its size might not be sufficient for increasingly advanced deep learning models. Despite that, Human3.6M is still frequently used due to its highly accurate 3D joint positions and diverse scenarios. However, as of now, it is no longer available through official sources.

3DPW

3D Poses in the Wild Dataset [46] is the first dataset truly outdoor ("in the wild") with accurate 3D poses for evaluation, taken from a moving phone camera. This novel approach allows for a more dynamic capture of human motion, especially in outdoor settings. With 60 video sequences and 18 3D models, it includes a variety of clothing variations and complex scenes. Its strengths lie in its ability to handle moving cameras, heading drift, occlusions, and multiple people visible in the video. However, its complexity could also be seen as a limitation, as it may require more advanced processing and analytical methods.

2 DESCRIPTION OF MULTIVERSE AND MULTIPOSE MODELS

The primary focus of this thesis is to enhance an existing model, *MultiPose*, that has been adapted from an original model, *Multiverse*, to predict human motion.

The original model, Multiverse, was first introduced in the article "The Garden of Forking Paths: Towards Multi-Future Trajectory Prediction" [24]. The original model's primary focus was the prediction of a human agent path within an environment using 2D video data. M. Cunico, in his master's thesis titled "Human Motion Anticipation through 3D Structured Multi-Future Trajectory Prediction" [24], subsequently modified this model to predict human body motion. The resulting model was then named MultiPose.

The motivation behind the selection of the Multiverse model as the base for the problem of human body motion prediction was motivated fundamentally by these considerations:

- *Generation of Multiple Outputs*: Multiverse has a multi-future capability thanks to its classifier component, which generates a heatmap of possible futures. This is particularly pertinent given the stochastic nature of human motion, where possible outcomes can vary substantially unless the motion in question is highly repetitive.
- *Semantic Context*: Furthermore, the Multiverse model has been designed to accept semantic segmentation maps as inputs, providing an opportunity to introduce semantics in the human motion context, the primary focus of this thesis.

2.1 FUNDAMENTALS

To help the explanation of the Multiverse and MultiPose models, some fundamental concepts used in the models will be introduced in the following sections.

Supervised and unsupervised learning

Among the different approaches to machine learning, supervised and unsupervised learning represent two fundamental approaches.

- *Supervised Learning*: the model learns from labeled training data, where the desired output is known. This approach can be visualized as a teacher-supervised learning process where the model under guidance. The aim is to learn a function that best maps the input data (features) to the corresponding output. Supervised learning is typically used for tasks such as classification (where the output is a categorical variable) and regression (where the output is a continuous variable).

- *Unsupervised Learning*: Unlike supervised learning, unsupervised learning involves training the model on data without predefined labels, i.e., the desired output is unknown. Here, the model's objective is to identify patterns, structures, or relationships within the input data. This is useful for clustering data into different groups, detecting outliers, or reducing the dimensionality of the data for easier visualization or computation.

The supervised learning approach is used both for the classifier and the regressor that constitute the Multiverse model. While the unsupervised learning will be used for Principal Component Analysis (PCA) for the dimensionality reduction task used in the preprocessing of the MultiPose model.

2.1.1 *Sequence-to-Sequence neural networks*

Sequence-to-sequence (Seq2Seq) models are a category of machine learning models primarily used for tasks that require the handling of sequential data. They were introduced for natural language processing (NLP), but Seq2Seq models have found use-cases across diverse applications such as machine translation [33], speech recognition [10], time series prediction [5], and image captioning [40].

At a high level, Seq2Seq models are composed of two main components: an encoder and a decoder. The encoder processes the input sequence, transforming it into a context vector which aims to encapsulate the information contained in the input sequence. The decoder then takes this context vector and generates the output sequence.

In essence, the model reads the input sequence, abstracts it into a fixed-length context vector, and then generates an output sequence from this context.

For example, in machine translation, the model has to translate a sentence from one language to another, so the lengths of the inputs may vary, and the inputs are usually different in size to the outputs. The input sequence (a sentence, for example) is encoded into a context vector, which the decoder uses to generate the output sequence that is the translated sentence.

Search

The generation of sequences in Seq2Seq models involves predicting a series of outputs, such as words or characters, one at a time. After each prediction, a decision must be made about which output to select before moving on to predict the next one in the sequence. This is essentially a search problem.

Ideally, the algorithm would have to search through all possible sequences to find the most probable one given the input and the model's learned parameters. But the complexity of this task grows exponentially with the length of the sequence, making efficient search strategies crucial for good performance. This topic will be better covered in the section Diverse beam search.

Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a class of artificial neural networks designed to recognize patterns in sequences of data, such as text, speech and time-series in general.

RNNs can remember the information computed in the previous steps, and this makes them suitable for dealing with sequential data. The "memory" in RNNs is captured through hidden states, which aim to encode the information of past inputs, which is crucial when an understanding of context over time is needed.

Seq2Seq Architecture

As mentioned, the seq2seq model basically consists of an encoder and a decoder:

- The encoder is typically implemented as a recurrent neural network (RNN) layer or a stack of RNN layers. It processes the input sequence one time step at a time and returns its own internal state. The outputs of the encoder are discarded, and only the final state is retained. This state serves as the "context" for the decoder.

- The decoder is also implemented as an RNN layer or a stack of RNN layers. It is trained to generate the next samples of the target sequence based on the context provided by the encoder's output and the previously generated samples from the target sequence. During training, it is conditioned on the previously generated samples and its task is to generate the next sample based on the context and the previous samples.

During training, a technique called "teacher forcing" can be used. It means that the decoder is fed with the correct previous samples as inputs. While this technique can provide benefits, it is important to notice that it can also create a discrepancy between training and inference. When using teacher forcing during training, the model learns to predict from the correct inputs at each step. However, during inference or real-world use, the model will rely on the predictions that it generates as inputs, which might be very different from the ground-truth that it was fed during training.

It is important to note that the side effects of teacher forcing can be mitigated by implementing approaches such as *Scheduled Sampling* [4] and *Professor Forcing* [22], but due to time constraints these methods have not been considered in this thesis.

2.1.2 *Search algorithms*

Diverse beam search has been used to produce qualitatively different outputs generated by neural networks, and it is also fundamental in the Multiverse and MultiPose models in order to produce multi-future predictions.

Greedy search

Greedy search is a simple and commonly used method for generating output sequences in Seq2Seq models. It works by always selecting the token (i.e., word, character, or feature) with the highest probability at each time step in the sequence, given the current context. However, this approach can lead to suboptimal solutions, as it fails to consider the overall sequence probability. Instead, it takes the locally optimal choice at each step, which can result in globally suboptimal solutions.

Beam search

Beam search alleviates some shortcomings of greedy search by maintaining a set of the most probable sequences (known as 'beams') at each time step, instead of just one. The width of the beam, i.e., the number of kept sequences, is a hyperparameter that governs the trade-off between computational complexity and output quality. By generating multiple output sequences, beam search provides a better chance of finding a globally optimal solution, which makes it particularly well-suited to the nature of Seq2Seq models where the goal is often to generate the most probable sequence given the input.

Diverse beam search

Standard beam search tends to generate a set of highly similar sequences given similar inputs, due to the shared beginning tokens among the beams. The lack of diversity among the top sequences may be a limitation, especially in tasks where multiple diverse solutions are required, as in the multi-future approach that is fundamental to this thesis.

To address this limitation, diverse beam search has been proposed [45]. The idea is to add a diversity-promoting term to the objective function during the search process. This term discourages the algorithm from selecting similar sequences for the top beams.

As a result, diverse beam search can provide multiple diverse and high-quality sequences, offering a broader range of possible solutions to choose from. This is particularly useful in tasks where more varied outputs are desired [17].

Diversity Strength

The *Diversity Strength*, in diverse beam search, is a parameter that controls the level of diversity in the generated results. It affects the trade-off between exploring diverse options and selecting high-scoring options.

The diversity strength term is used to encourage or discourage the selection of similar hypotheses, its impact can be qualitatively described as follows:

- A larger diversity strength term encourages the algorithm to prioritize diverse solutions. It discourages the selection of similar hypotheses and promotes exploration of different paths. This can lead to more varied and distinct results, but it may also sacrifice some quality in terms of the highest-scoring options.
- Reducing the diversity strength term allows similar hypotheses to have a higher chance of being selected. This can result in more focused and coherent outputs, as the algorithm is more likely to converge on high-scoring options. However, it may lead to less diversity in the generated results, with potentially fewer novel or alternative solutions explored.

Experimenting with different values of the diversity strength term is often necessary to find the desired level of diversity and quality for the particular application. A. Vijayakumar et al. [45] suggest a value between 0.2 and 0.8 for most applications and datasets.

2.1.3 *Graph attention networks*

Graph Attention Networks (GATs) are a class of machine learning algorithm specifically designed to handle data structured as graphs. In a graph, data points (referred to as nodes) are connected via relationships (edges), forming a complex network. GATs, using an attention mechanism, determine the importance of each connection in this network as part of the learning process.

When data is represented in a graph, each data point (node) is associated with others through various relationships (edges). This allows the GAT to capture both the characteristics of the node itself and the context given by its relationships with other nodes in the graph. In GATs, the importance (or weight) given to each relationship (edge) is calculated by the attention mechanism, offering an adaptive learning:

this enables the network to capture complex and non-linear relationships within the graph structure.

The edge function in GATs determines the attention weights, which are a score representing the importance of one node to another. This function can adopt various forms. For instance, it could be a multilayer perceptron (MLP), a type of neural network that can model non-linear relationships. With an MLP as an edge function, attention weights are derived from a non-linear function of the nodes' characteristics, offering a more flexible and adaptable modeling approach.

2.2 ORIGINAL MULTIVERSE MODEL

The neural network model central to this thesis was introduced in the paper titled "The Garden of Forking Paths: Towards Multi-Future Trajectory Prediction" [24]. In the following sections, a description of the model will be provided, while also presenting an updated version called "MultiPose" [12], which is derived from the original Multiverse framework, but is adapted to predict human body motion.

2.2.1 *Overview of the model*

The Multiverse model focuses on predicting multiple plausible future trajectories of individuals navigating (outdoor) environments based on visual inputs from cameras (typically security cameras). Basically, the objective is to predict future paths of a single human agent, given the past video frames, location of the agent in the scene and a semantic segmentation map. By utilizing a particular method of location encoding and convolutional RNNs (inspired in part by object-detection methods [39]), the model can generate diverse plausible future paths.

2.2.2 *Architecture*

The Multiverse model can be seen as a variant of the sequence-to-sequence (seq2seq) structure. It includes an encoder, the History Encoder, which processes the past sequence of frames and locations, and two decoders, the Coarse and Fine Location Decoders, which generate the sequence of predicted locations.

As illustrated in the image 1, the model's output is a fusion of the outputs generated by a classifier, referred to as the coarse location decoder, and a regressor, which is the fine location decoder. This characteristic enables the model to generate multiple future trajectories. The classifier provides a heatmap of potential future locations, while a search algorithm uses this probability distribution across the image to generate coarse trajectories. These trajectories are then refined thanks to the fine location decoder.

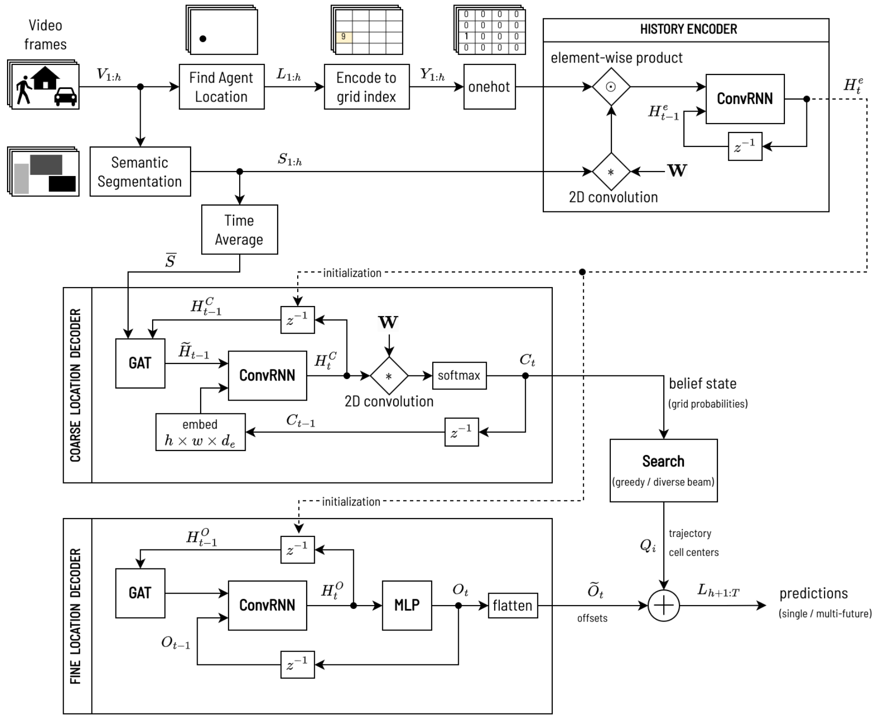


Figure 1: Multiverse block diagram.

History Encoder

The History Encoder works by encoding the past sequence of agent locations and video frames. Each agent location is encoded as a cell in a multi-scale grid, and each video frame has a semantic segmentation map associated to it. The segmentation map is produced during the preprocessing of the video frames thanks to a semantic segmentation model, which in this case is the Deeplab [9], which produced a total of 13 semantic classes for the image pixels. Note that in the block diagram 1, only one grid is considered for simplicity.

These locations and semantic maps are then fed into a Convolutional recursive neural network (ConvRNN), which generates a sequence of spatio-temporal feature maps. This sequence of maps constitutes the output of the History Encoder and serves to initialize the decoders. While the hidden state of the encoder is randomly initialized.

In particular, the semantic segmentation map is processed by a 2d convolution layer, while the ConvRNN is constituted by a ConvLSTM (convolutional long short-term memory) layer.

Coarse Location Decoder

The Coarse Location Decoder operates as a grid cell classifier. It is constituted by a ConvRNN which hidden state is initialized by the History Encoder. The ConvRNN takes two inputs:

- $\tilde{H}^C_{t-1}$: the past hidden state (H^C_{t-1}) is processed by a Graph Attention Network (GAT), which models the spatial dependencies on the grid, ensuring more realistic location predictions. This is works thanks to the notion that the trajectory is most likely going to spread through adjacent cells, instead of jumping to distant cells.

- $\text{embed}(C_{t-1})$: the hidden state is also processed by a 2D convolutional layer, and a $\text{softmax}()$ operation to obtain a belief state C_t which represents the "heatmap", or the probability distribution generated by the classifier over the grid.

The result is a "belief state", a probability distribution over grid cells, which represents the decoder's predicted locations at a coarse level.

The encoder hidden state update is represented by the Equation 1 [24].

$$H^C_t = \text{ConvRNN}\left(\text{GAT}\left(H^C_{t-1}\right), \text{embed}\left(C_{t-1}\right)\right) \tag{1}$$

Fine Location Decoder

The Fine Location Decoder operates as a regressor, which refines the coarse predictions of the Coarse Location Decoder. For each grid cell an offset is computed by a ConvRNN updated with a GAT as in the coarse decoder. The offset vectors are then computed by a multilayer perceptron (MLP).

The inputs to the ConvRNN in this case are both the fine decoder hidden state processed by a GAT, and also the offsets vectors (output of the MLP). The hidden state update function can be seen in Equation 2.

$$H^O_t = \text{ConvRNN}\left(\text{GAT}\left(H^O_{t-1}\right), O_{t-1}\right) \in \mathbb{R}^{H \times W \times d_{\text{dec}}} \tag{2}$$

Where H^O_{t-1} is the hidden state of the fine decoder (O stands for offsets), and O_{t-1} are the grid cell offsets.

Prediction Generation

The final prediction combines the outputs of the Coarse and Fine Location Decoders.

The Coarse Location Decoder identifies the most probable grid cells, and the Fine Location Decoder provides the more precise location within each cell.

The path selection process depends on the type of prediction needed, which in turn depends on the type of search algorithm used.

- For single-future predictions, for each future time-step the grid cell with the highest probability is chosen from the coarse prediction (greedy search)
- For multi-future predictions, a set of potential grid cells are chosen from the coarse prediction by applying a diverse beam search (as seen in Section 2.1.2.3).

In conclusion, this architecture leverages the seq2seq paradigm to effectively process past information (via the History Encoder) and generate future predictions (via the two decoders). Its unique feature is its combination of classification (coarse decoder) and regression (fine decoder) mechanisms to generate both diverse (multi-future) and precise predictions.

2.2.3 *Description of loss function*

The model loss is a linear combination of the classification and regression losses. As in the paper that first introduced Multiverse [24], the losses will be described below.

Coarse location decoder loss

For the coarse decoder, the cross-entropy loss is used, and the formula is shown in Equation 3. This loss function is relative to the classification task of the coarse locator.

$$\mathcal{L}_{cls} = -\frac{1}{T} \sum_{t=h+1}^{T} \sum_{i \in G} C^*_{ti} \log (C_{ti}) \tag{3}$$

Cross-entropy is a favored loss function for machine learning classifiers due to its ability to quantify the difference between predicted probabilities and true class labels. Derived from information theory, it offers a probabilistic interpretation by measuring the average information required to determine the correct class. As a differentiable function, it facilitates gradient-based optimization methods, enabling efficient parameter adjustments through backpropagation.

By minimizing cross-entropy loss, the classifier aligns with maximum likelihood estimation principles, maximizing the likelihood of observed data.

Fine location decoder loss

For the fine location decoder, a smoothed L_1 loss is used:

$$\mathcal{L}_{reg} = \frac{1}{T} \sum_{t=h+1}^{T} \sum_{i \in G} \text{smooth}_{L_1} ((L^*_i - Q_i), O_{ti}) \tag{4}$$

Where O_{ti} is the offset relative to the grid cell i at the time t, L_t^* is the ground truth location and Q_i is the location of the $i-th$ cell grid center.

The smoothed L1 loss is utilized, by inspiration from object detection methods, for several reasons [24].

- It offers a more robust alternative to the standard L_1 loss by incorporating a smooth transition around the origin. This smoothness property helps to reduce the impact of outliers and noisy data points, making the loss function less sensitive to extreme errors.
- Moreover, the smoothness property of the loss function ensures that small errors in the predictions contribute less to the overall loss, providing stability during training. This can prevent the model from overfitting to individual outliers or noisy examples.

2.3 MULTIPOSE: HUMAN MOTION PREDICTION

As described in Section 2.2, the Multiverse model works with 2D video data. In order to achieve the objective of human motion prediction, the model instead has to work with 3D data, specifically with the joint values of a simplified human skeleton model. To this end, Cunico adapted the model by modifying the preprocessing of the joint trajectories and fitting a prediction model for each joint. All the models predictions are then combined using a beam search, with a technique similar to the one seen in the Multiverse model [24], Diverse beam search.

2.3.1 *Human motion prediction*

It is first necessary to introduce what we mean by human motion prediction and how the motion data is represented.

Human Body Representation

In the context of representing human poses, various approaches exist using kinematic models, which provide simplified representations of the underlying skeleton. One commonly used model is the OpenPose model [6], which has gained significant popularity due to its robustness and versatility in estimating human pose. Additionally, other skeleton models, such as the International Society of Biomechanics (ISB) biomechanical skeleton model [47], are also utilized.

In this thesis in particular, the Skinned Multi-Person Linear Model (SMPL) model is used [27] since the AMASS dataset is used [28], but the methodologies are applicable independently of the body representation choice.

Other benefits of using the SMPL model are that it employs an overall more natural structure of the kinematic skeleton rather than the one chose by H3.6M [20], and not as simplicistic as the one in OpenPose [6]. Another advantage of using SMPL is that it can be amplified by adding further body joints, such as realistic hands movements given by the addition of the MANO [41] model; furthermore, and SMPL-X [34] adds facial expression features that could be crucial in further research on the understanding of semantic features of human subjects. On top of all of that, visualization tools [34] exist for the SMPL model, which can produce realistic 3D full-body models with many adjustable parameters to vary the body shapes, this can be useful both for visual validation of results, but also to check for accurate collisions in the body (which cannot be done as accurately with a simple kinematic skeleton model).

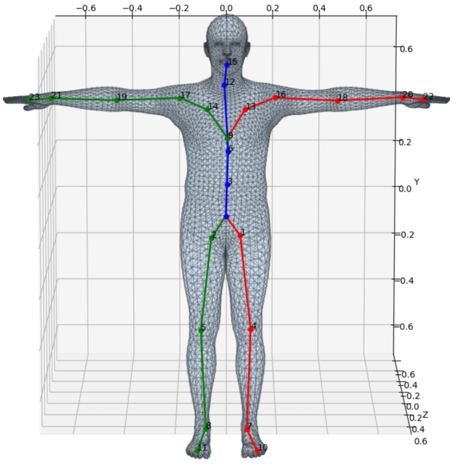


Figure 2: SMPL human skeleton model. [18]

These models consist of interconnected segments representing different body parts and joints, enabling the estimation and tracking of joint angles during motion analysis. By employing these skeleton models, researchers can efficiently process and analyze human motion data, facilitating various applications in fields such as animation, biomechanics, and human-computer interaction.

Human Motion

Once a simplified skeleton representation is available, human motion can be described as a time series of joint angle values. These values capture the relative positions and orientations of the body joints, providing a comprehensive representation of the subject's movements. Various techniques can be employed to capture human motion, including Inertial Measurement Units (IMUs) and reflective markers placed on key body landmarks. These techniques enable the recording of precise joint angle measurements over time, allowing for detailed analysis and understanding of human movement patterns.

The rotational information of the joints can be represented in different ways:

- Euler angles: Euler angles are easy to comprehend for humans and can provide an intuitive representation of 3D rotations. However, they suffer from the issue of singularities, which can lead to ambiguities and numerical instability in certain orientations.
- Quaternions: Quaternions offer a singularity-free representation of 3D rotations and can be efficiently interpolated. However, they require normalization to maintain their mathematical properties, which adds computational overhead and necessitates special treatment when integrating them into a neural network model.
- Exponential map / angle axis: The exponential map or angle-axis representation mitigates the singularity problem of Euler angles and avoids the normalization requirement of quaternions. It provides a compact representation with three values, facilitating ease of use. However, its interpretation may be less intuitive for humans compared to Euler angles.
- Rotation matrices: Rotation matrices provide a complete and singularity-free representation of 3D rotations. They are widely used in graphics and geometry computations. However, their drawback lies in the high dimensionality, requiring nine elements to represent a 3D rotation, which can increase computational complexity and memory usage.

Human Motion Prediction

In the context of motion prediction, the objective is to generate possible future "frames" of human motion based on the past motion data. Given the historical trajectory of joint angles, a predictive model is employed to generate updated values for future time points. The prediction model leverages the temporal patterns and dependencies observed in the past motion data to forecast the future joint angles.

2.3.2 *Dimensionality reduction*

Two options were considered by Cunico [12] to adapt the Multiverse model for the task of human motion prediction.

1. Increasing the input size to 3-dimensional, and consequently increasing all the layer dimensions. This proved inefficient both computationally (the training / inferencing times increased exponentially), and in the sense of training effectiveness. The latter being a result of the neural network architecture, where sparse

data meant a more difficult learning target for the model; particularly for neural networks which implement convolutional layers and must be treated carefully [26].

2. Reducing the dimensionality of the input: this in turn can be done with multiple techniques, for instance by projecting 3D trajectories onto a pre-defined 2D plane (e.g., the plane determined by the first two parameters) or reducing the dimensionality by simply removing one component from the rotational information. That approach may not be optimal in terms of information loss, for this reason in the Multipose work of Cunico [12] he chose to implement a principal component analysis to project the 3D rotations in a 2D space with the minimum information loss.

The transformation process involves two steps: applying dimensionality reduction to reduce the necessary parameters from 3 to 2, and subsequently scaling the resulting 2D trajectories to fit within the Multiverse-interpretable video scene ($V_H \times V_W$ in pixels).

PCA

Principal Component Analysis (PCA) is a technique used for dimensionality reduction to preserve as much variability (or information) in the data as possible.

PCA works by finding a new set of orthogonal axes, the so-called principal components, that are linear combinations of the original axes and capture the most variance in the data. The first principal component captures the most variance, the second principal component captures the second most variance, and so on.

The steps to perform the PCA and dimensionality reduction are as follows:

1. *Standardization*: Normalize the data by subtracting the mean and dividing by the standard deviation of each feature.

2. *Covariance Matrix*: Compute the covariance matrix of the standardized data to capture the variance and correlation structure. The covariance between two variables indicates how they vary together, for instance, large covariance indicates strong correlations, while small or zero covariance values indicate little to no linear relationship.

3. *Eigendecomposition*: Perform eigendecomposition of the covariance matrix to obtain the eigenvectors (principal components) and eigenvalues.

4. *Dimensionality Reduction*: Select the top eigenvectors corresponding to the largest eigenvalues to form a projection matrix. Project

the original data onto the projection matrix to obtain the reduced-dimensional representation.

2.3.3 *Range Saturation and Prediction Combination*

At this stage, it is assumed that the models for each joint have generated trajectory predictions (single or multi-future), and it is possible to apply the inverse transformation of dimensionality reduction to return to the 3D representation of rotations.

Range Saturation

Given the absence of explicit constraints on the output ranges of the prediction models, it is possible for the generated trajectories to contain rotations that exceed the natural limits of human body joints. While a well-trained model should ideally learn to avoid such out-of-bounds predictions, it is necessary to apply range saturation to address this issue and ensure that the predicted rotation values fall within acceptable boundaries. By imposing saturation, any predicted rotations beyond the permitted range can be clipped according to the known constraints.

The conversion from axis-angle representations to Euler angles is performed to comply with the kinematic rotation constraints of the human body. A technique similar to the one adopted by OpenSim [13] is used by Cunico in the Multipose model [12] to saturate any predicted values exceeding the predefined limits.

These limits are statically defined as ranges of allowed rotations. The rotation ranges are empirically determined based on the maximum and minimum rotation values observed in the joint rotations of the dataset and kinematic model, specifically the SMPL skeleton in the AMASS dataset [27, 28].

Prediction Combination

Since multiple models are used to generate independent trajectories for each joint, it is necessary to combine these trajectories to obtain a comprehensive human body motion.

Each model predicts a number N of future scenarios and assigns a probability (that quantifies the model belief state) to each trajectory. Considering J joints, with N trajectories per joint, an efficient approach is to apply a beam search to generate comprehensive trajectories. However, unlike the beam search performed in Multiverse, the search process advances from joint to joint instead of progressing in time steps at each iteration.

2.4 MODIFICATIONS TO THE MULTIPOSE MODEL

While the main focus of the thesis is the implementation of semantic information, other aspects of the Multipose model were changed. The most relevant changes will be discussed in the following sections.

2.4.1 *Preprocessing*

Multiple framerate support

An added functionality to the model preprocessing is the ability to handle motion capture (mocap) data with different source frame rate. This is useful because in mocap datasets like AMASS [28], that is a group of mocap from various datasets, the frequency of the data samples may be different.

Downsampling

Moreover, the target frame rate (the one which the model should work with) was assumed to be an integer multiple of a fixed downsampling factor. This meant less flexibility in the possible mocap that could be used.

Due to the constant frame rate assumption, the downsampling method was a simple decimation, but has now been updated with a more robust SLERP (spherical linear interpolation).

2.4.2 *Training*

Working frame rate increase

The original Multipose model working frequency was set to 10Hz but has been increased to 25Hz to better capture dynamics in the human joint trajectories.

This means that by keeping the same temporal time-frame, the input and output sequences have more frames to work with.

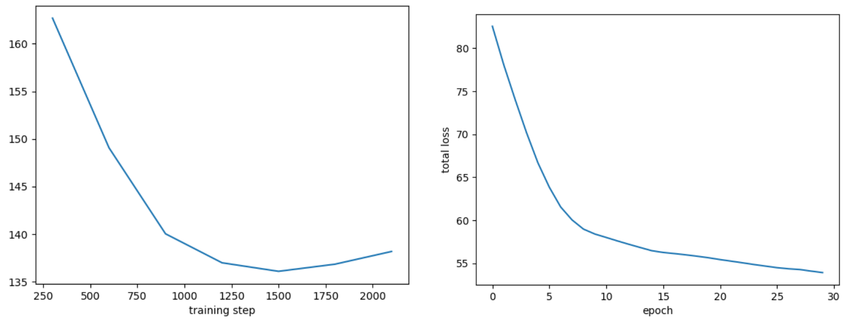
This was done because it is generally acceptable to increase the input and output sizes of a sequence-to-sequence model, as long as the increased computational requirements, such as memory and processing power, are available.

Training statistics visualization

The evaluation on the validation set is not synchronous with the epochs, so the x-axis unit of measure of the plot in Figure 3a is *train-*

ing steps, where the total number of training steps is defined as in Equation 5.

$$\text{total\_steps} = \text{epochs} \cdot \frac{\text{total\_samples}}{\text{batch\_size}} \tag{5}$$



(a) Evaluation during trainig and total loss for the "pelvis" joint, which is the global orientation of the body.

(b) Evaluation and total loss for .

Figure 3: Training statistics visualization.

The plots of the loss function and evaluation metric in the training process provide insights for model analysis. In particular, in Figure 3b the total loss is shown, since it is a combination of multiple losses, see Section 2.2.3. The loss function plot indicates model convergence and overfitting, with a decreasing trend reflecting how well the model is converging during training. Deviations or increases in the validation set loss may indicate overfitting, so that we have an indication of when to stop the training.

2.4.3 *Post-processing*

The main contributions in the post-processing of the Multiverse model are the following:

- Due to an error in the trajectory saturation, the allowed ranges for joint rotations were not applied correctly.
- The whole process of pre-processing, training, inference and metric evaluation has been streamlined for a more efficient way of defining experiments with the ability to change hyperparameters and mocap datasets.

3 IMPLEMENTATION OF SEMANTICS IN THE MULTIPOSE MODEL

3.1 DATASETS

The dataset that has been chosen for human body motion data is AMASS (briefly described in section AMASS). Although Human 3.6 Million (H3.6M) is currently one of the most used dataset in the field of human motion prediction, the main advantages of AMASS are:

- AMASS size is considerably larger than Human 3.6 M, comprised of many independent datasets unified through the SMPL model. This means that there is more variety in types of actions and number of mocap hours necessary to train deep learning algorithms.
- The BABEL dataset [37] that complements AMASS with semantic information is crucial for the task at hand. While BABEL does not cover the semantic information of all the AMASS mocaps, it still represents a clear upgrade to Human 3.6 M.

Although the techniques described in this thesis for the implementation of semantic information in a human motion prediction model can be applied generally, the specific techniques were adapted specifically to work with the BABEL dataset.

For this reason, a concise description of the BABEL dataset will be provided in the following sections.

3.1.1 *BABEL dataset*

BABEL is a comprehensive dataset developed to facilitate an in-depth understanding of the semantics of human movement. BABEL focuses on providing dense action labels for high-quality motion capture (mocap) sequences. This involves categorizing around 43 hours of mocap sequences from the AMASS dataset with two levels of action abstraction: sequence labels and frame labels.

Semantic data

To construct BABEL, motion-captured sequences from AMASS were rendered into videos and shown to human annotators, who assigned action labels.

Annotations occurred at two resolution levels:

1. *Sequence labels*: describing the overall action being performed. This is similar to a summarization of the overall action. For instance, a sequence of a person playing basketball would be broadly labeled as "Playing basketball".

2. *Frame labels*: providing a lower level description of the single body actions occurring in specific time windows. For example, within a basketball-playing sequence, frames labels might be labeled as "running", "shooting", "jumping", capturing the range of actions involved in the higher level activity of playing basketball. Frame labels are aligned precisely with their corresponding frames in the sequence, and often overlap to capture simultaneous actions.

BABEL's sequence labels provide a broader view of the sequence, as one might call "high-level description of the whole sequence" - but they are not limited to a single action category per sequence, as a whole sequence could be described as "play sport" and "interact with object". This is because human actions are often complex and involve more than just a single activity.

Label Processing and Semantic Categories

Action categories: the human annotators write "raw" action labels that need to be clustered into 260 distinct action categories, which occurrence graph is partially shown in Figure 4. For example, raw action labels for the category "walk" could be: walking, walk forward, walk straight, walk normally etc.

This step of clustering and standardization of action description is critical for the machine learning objective. But it should be noted that the authors of BABEL themselves admitted that due to the long-tailed distribution of the 260 action categories render the dataset a not trivial choice to for action-recognition machine learning algorithms training.

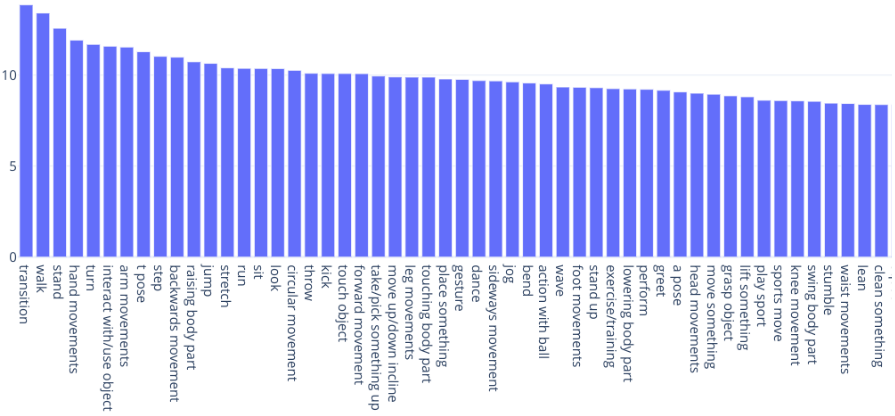


Figure 4: BABEL categories $\log_2$ of number of occurrences in the dataset (partial image). [37]

Action macro-categories: Furthermore, these action categories were further grouped into eight semantic *macro-categories*.

The macro-categories are described below [37] and a chart of the distribution of occurrences in the BABEL dataset is shown in Figure 5:

1. *Simple dynamic actions*: low level atomic actions such as "walk and jump".
2. *Static actions*: when the body pose is mostly still, with actions such as "lean" and "sit".
3. *Object interaction*: when the human agent is interacting with an inanimate object, such as "place something" and "grasp object".
4. *Body part interaction*: when a body part of the actor is coming in contact with another part of the actor's body, such as "touching face", "scratch".
5. *Body part*: this groups all the actions specific to a distinct body part such as "arm movements" and "head movements".
6. *Type of movement*: it refers mostly to a movement that can be described by an adjective such as "circular movement" and "backwards movement".
7. *Activity*: the movement is described by a high-level activity like "play sport" and "dance".
8. *Abstract actions*: although there are not many abstract actions in the dataset, this macro-category groups categories such as "excite" and "support".

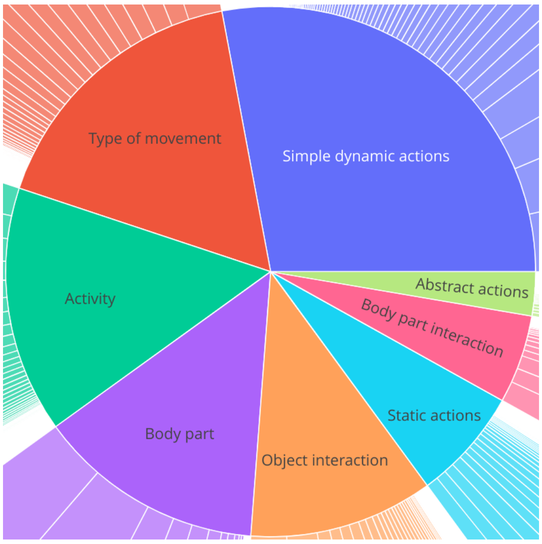


Figure 5: BABEL macro-categories occurrence distribution in the dataset. [37]

3.1.2 *AMASS subsets*

The BABEL dataset supplements the AMASS mocap dataset by providing semantic context to the movements. However, it is important to note that the distribution of this semantic information across different subsets of AMASS is not equal. The number of processed annotations varies across different subsets, as does the total number of frame and sequence annotations.

Given that our experiments do not utilize the entirety of AMASS, selecting the most relevant subset for our problem is crucial. This choice is influenced by how many mocaps have semantic annotation. As we aim to provide a high-level semantic description of the scene, sequence labels are more suitable. Moreover, most datasets contain a higher number of sequence labels compared to frame labels as can be seen in the Table 1.

The subset finally selected is BMLrub. This subset not only offers a considerable volume of mocap data (over 500 minutes), ranking it among the most populated datasets in AMASS, but it also has a good semantic coverage compared to other subsets, as shown in Table 1. BMLrub, designed with action recognition and other objectives in mind, includes a diverse range of actions, making it a suitable choice for our experiment.

Table 1: Semantic information for some AMASS subsets. "anno. %" indicates the percent of mocaps with sequence or frame labels over the total in the subset. EKUT, KIT [29]; SSM [28]; BMLmovi [15]; BMLrub [44]; CMU [7].

Mocaps / Datasets	**EKUT**	**KIT**	**SSM**	**BMLmovi**	**BMLrub**	**CMU**
Total number	349	4232	30	1887	3061	2088
In BABEL	265	3283	24	1378	2736	1702
Sequence anno. %	60.2	59.0	63.3	57.2	67.4	60.3
Frame anno. %	32.4	41.2	26.7	29.9	29.6	47.9

3.2 ADAPTING THE SEMANTIC MAP OF MULTIPOSE

To state the problem at hand one more time: the objective is to augment the predictive capabilities of a neural network model, by adding semantic information. The model is MultiPose, or Multiverse in its core, and so we need to consider how the semantics work in these models.

3.2.1 *Multiverse semantics*

Originally, the construction of the semantic map was intended to allow Multiverse to establish correspondences of objects within the scene. In that context, the semantic segmentation map represents a 2D scene as captured by a camera.

However, in our particular application, the scene does not contain spatial information. The "image" in Multipose is essentially blank; the information is conveyed through the "agent's" location (as in the terminology from Multiverse).

What was once the "agent" location on screen for Multiverse, is now joint angle coordinates in the 2D dimensionally reduced space of Multipose.

Therefore, semantics intended as action category does not correspond directly to any visual component in the scene. With that in mind, in the following sections (3.2.2), two approaches for action semantic representation compatible with the Multiverse models are described.

Semantic segmentation map internal representation:

the semantic segmentation map that the Multiverse preprocessing pipeline receives as input is a tensor of shape:

$$shape(\text{SegMap}_{raw}) = (T, S_H, S_W) \tag{6}$$

Where T is the number of video frames the semantic segmentation map is computed for; S_H and S_W are respectively the height and the width of the scene in pixels (which is generally a lower resolution than the actual video dimensions);

This "raw" representation generated by the semantic segmentation model is then converted into one-hot encoded matrices representing a binary mask of the features in the image.

$$\begin{aligned} &shape(\text{SegMap}_{proc}) = \\ &= shape(one\,hot(\text{SegMap}_{raw}))) = (T, S_H, S_W, K) \end{aligned} \tag{7}$$

Where K is the number of semantic classes and the datatype of the elements in the tensor is boolean.

3.2.2 *Semantic representation approaches*

Bitwise multi-class

Assuming that the class information for each pixel in the scene is internally represented as an N-bit unsigned integer variable, there can

N possible semantic action classes that can be represented independently. By representing the mocap action semantics as a bit in an N-bit number, the simultaneous presence of multiple action categories can be dealt with.

But it is important to note that Multiverse performs a one-hot encoding of the semantic classes of the scene, generating a binary mask (the size of the scene) for each class.

There are two main issues when assuming N (independently activatable) action categories for describing human motion:

1. The potential number of classes seen by Multiverse is enormous. Considering that for each Multiverse class, a matrix is created with a size equal to the scene: $S_H \times S_W$ pixels, the total number of bits required would be $S_H \cdot S_W \cdot K$, as shown in Equation 3.2.1.1. With $N = 64$, we would have approximately $2^N \simeq 18 \cdot 10^{18}$ Multiverse semantic classes and approximately 10^{22} bits of memory usage for a single mocap sub-sequence in the worst case. Such a memory requirement is not physically feasible.

   Although the number of possible permutations of action classes would not reach that extreme number from the AMASS dataset possible combination of action classes, it would still grow rapidly to an unmanageable number.

2. The presence of multiple action classes simultaneously results in entirely different semantic classes from the perspective of Multiverse, owing to the one-hot encoding of the scenes. This would lead to a long-tailed distribution of semantic classes (where the most of the classes would appear only once per dataset) and lead to an ineffective learning by the model.

Fixed single-class

To overcome the excessive number of classes of the previous approach (Bitwise multi-class), a single semantic class of action will be assigned to each mocap sequence.

This means that once the action category is defined for the motion sequence, it is applied uniformly to the semantic segmentation map of the Multiverse model.

The method of choosing which label should represent an action sequence is discussed in Section 3.3.

The main point is that, as stated in the introduction 3.2.1, the semantics employed in the Multiverse model work with spatial attributes and visual objects in a scene. On the other hand, the semantics that we want to implement describe a more abstract concept of semantics, in the sense that it is not tied to the physical image, nor to the trajectories of the body joints. For this reason, the action semantics will be represented in the semantic segmentation map as a full image with one class. Due to Multiverse performing one-hot encoding on the semantic classes, the resulting action semantic information is represented as described hereafter.

Let I denote the original image with dimensions $S_H \times S_W$, where each pixel $I_{i,j}$ takes a value between 1 and K, representing the semantic class of that pixel.

In our scenario, the entire image is a solid color, indicating that if one pixel has the value k (representing class j), all other pixels in the image will also have the same value k.

The one-hot encoding process generates K binary matrices for each frame, each with the same dimensions as the original image. These matrices will be denoted as M_k for $k = 1, 2, \dots, K$ so that M_k represents a binary mask of the semantic segmentation map for the class k.

Therefore, the one-hot encoding transformation can be defined as follows:

$$M_k(i,j) = \begin{cases} 1, & \text{if } I_{i,j} = k \\ 0, & \text{otherwise} \end{cases}$$

Where $i = 1, 2, \dots, S_H$ and $j = 1, 2, \dots, S_W$. By applying this transformation, we obtain K binary matrices $M_1, M_2, \dots, M_K$, where each matrix M_j represents the j-th action semantic class.

Each matrix is populated with zeros, except for the matrix associated with the semantic class of the scene.

This is the approach we chose to implement the semantic in the MultiPose model.

3.3 ACTION LABELING APPROACHES

The technique employed to incorporate semantic data into the model partially depends on the format in which this semantic data is presented, but can in theory be applied to a wider rage of cases, independently of the dataset choice.

In the following sections, three separate approaches will be discussed, about the representation of semantic data in the context of human body motion, with the third approach ultimately being the chosen one (3.3.3).

3.3.1 *Raw Sequence Labels*

Initially, an attempt was made to mimic the clustering methodology already implemented by the BABEL dataset, but instead of creating new classes names, the classes themselves would be the raw label words.

This approach involved extracting individual words from raw labels associated with each mocap sequence and then transforming these words into a one-hot encoded vector encompassing more than 500 classes (raw words) to represent semantics.

To reduce the word count, a random forest model was trained using both the one-hot encoded semantic vectors and the kinematic information of the human motion.

Subsequently, the feature importances of these raw words were extracted from the model, intending to retain a lower number of features, so keeping only the words that ranked as the most valuable predictors.

3.3.2 *Action Categories*

Despite its initial promise, the raw word approach encountered several issues. For example, similar words such as "walk" and "walking" were classified differently, adding unnecessary complexity to the model. To resolve this, the focus turned to using the preprocessed action categories of BABEL, which offer a more structured and reliable grouping of semantic action classes.

However, this approach of using "processed categories" presented another set of challenges. Given that each mocap sequence was associated with multiple action categories, it is difficult to assign a primary category to represent the sequence semantically.

Even though 260 categories was a marked improvement over the Raw Sequence Labels approach, it was still an excessive number of categories compared to the one found in the Multiverse semantics (History Encoder).

3.3.3 *Macro-Categories*

Since the use of all 260 action categories in BABEL was still an excessive number of classes to be handled by the Multiverse model, we considered using a further clustering of action classes already made available by BABEL [37]. This clustering individuates 8 macrocategories into which to group all the 260 categories, which is a much more manageable number of semantic classes for our model.

The original Multiverse model utilized a semantic segmentation map, which comprised 13 semantic classes. This is a number that more closely aligns with the nine macro-categories derived from the eight clustered categories from BABEL plus an additional category representing unknown actions.

Since the semantic information relative to actor actions is presumed to be "high-level", we can assume an unlikely scenario to be able to obtain a detailed frame-by-frame description of the actions. Instead, it is more plausible to achieve a higher-level description of the action being performed by the human agent, especially in an industrial environment where worker tasks are often scheduled, periodic, and typically limited in variety.

This led to the final decision to implement this "macro-category" approach in the final model.

3.4 MOCAP LABELING

Based on the considerations presented in the previous sections, the two approaches we will follow are summarized below:

1. **Semantic data**: we will utilize BABEL macrocategories of sequence labels, which refer to the high-level categories that describe the entire sequence.
2. **Multiverse implementation**: the selected method is the one described in Fixed single-class, where a mocap sequence is described by a single semantic class.

Now, it is necessary to define how to assign a specific class to each mocap sequence. This is because each sequence is typically described by a list of categories, but with the described implementation approach, it is possible to assign only a single semantic class to each mocap.

The chosen mapping is based on the dominant macrocategory in the sequence, which is found based on the sum of "importance" of each category. So if a sequence is semantically represented by a list of categories (in italics) like such:

$$\text{sequence} = [\textit{walk}, \textit{play sport}, \textit{jump}]$$

each category is grouped into its belonging macrocategory (in bold), and to each category is assigned an importance score:

$$\text{sequence} = \begin{bmatrix} \textbf{Simple Dynamic Actions} : \begin{Bmatrix} \textit{walk} = 7 \\ \textit{jump} = 9 \end{Bmatrix} \\ \textbf{Activity} : \{\textit{play sport} = 10\} \end{bmatrix}$$

Then the macrocategory importance is given by the sum of the categories importances:

$$\text{sequence} = \begin{bmatrix} \textbf{Simple Dynamic Actions} = 16 \\ \textbf{Activity} = 10 \end{bmatrix}$$

Yielding a total importance score of 16 for the macrocategory *Simple Dynamic Actions* and 10 for the macrocategory *Activity*. This means that the resulting semantic class which the sequence will be labeled with the macrocategory "*Simple Dynamic Actions*".

To formalize mathematically that example in the context of our implementation, consider the 260 action categories in BABEL (Label Processing and Semantic Categories) denoted as $c_1, c_2, c_3, \dots, c_{260}$. Assume there exists a function f that maps each category c_j to its corresponding macro-category C_k and a function g that assigns an importance value v_j to each category (see Section 3.4.1);

$$f : c_j \mapsto C_k \quad g : c_j \mapsto v_j$$

where $j \in \{1, 2, \dots, 260\}$ is the index of the category in BABEL and $k \in \{1, 2, \dots, 9\}$ is the index of the macrocategory among the ones highlighted in Label Processing and Semantic Categories, note that in the list of BABEL macrocategories only 8 appear, but we consider 9 macrocategories since we added the "unknown" category to handle motions without semantics.

A mocap sequence S can be semantically represented as a list of categories:

$$S_{\text{cat}} = \{c_i, \dots, c_n\}$$

$$S_{\text{macro cat}} = S_{\text{cat}} \xrightarrow{f} \{C_1, \dots, C_n\} \tag{8}$$

$$S_{\text{importances}} = S_{\text{cat}} \xrightarrow{g} \{v_1, \dots, v_n\} \tag{9}$$

where:

- S_{cat} is the list of action categories that represent the semantics in the sequence S.
- $S_{\text{macro cat}}$ is a list of the macrocategories corresponding to the categories in the list S_{cat}.
- $S_{\text{importances}}$ is the list of importance score for each category in the list S_{cat}.

The number of elements in $S_{\text{macro cat}}$ is the same as in S_{cat}, but while every element in S_{cat} is unique, in $S_{\text{macro cat}}$ the elements may repeat since the number of macrocategories is lower than the number of categories.

We now one-hot the macrocategories list $S_{\text{macro cat}}$, for each macro-category k, assuming the sequence is described by N categories:

$$S_{OH}(k) = \text{one-hot}(S_{\text{macro cat}}, k) =$$

$$\{b_1, \dots, b_i, \dots, b_N\} \quad \text{where} \quad b_i = \begin{cases} 1, & \text{if } C_i = k \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

The final step is to compute the elementwise product ($\odot$) between $S_{\text{one-hot MC}}$ and the importance scores list $S_{\text{importances}}$ and the elements of the resulting list to obtain the importance of the macrocategory:

$$S_{\text{macro score}}(k) = \sum_{i=1}^{N} S_{\text{OH}}(k) \odot S_{\text{importances}} \quad (11)$$

The dominant macrocategory, so the one which will label the whole sequence, is the macrocategory with the highest importance score:

$$C_{\text{dominant}} = max(S_{\text{macro score}}) \quad (12)$$

In conclusion, the labeling of a mocap sequence is done by evaluating the importance of the categories from which it is described and keeping the macrocategory with the highest importance. Now a fundamental problem remains, which is: how do we determine the "importance" of a category? This topic is discussed in the Section 3.4.1.

3.4.1 *Category importance*

The last step for introducing the action semantics in the model is to define a way to assign the importance of a category in describing the semantics of a whole motion sequence.

Two approaches have been considered:

1. *Predictive importance*: one approach involves training a random forest model (like the one described in Raw Sequence Labels) to obtain a feature importance list. By training the random forest model on both kinematic and semantic data, with motion sequence prediction as the target, we assess the predictive performance of each feature, including the semantic class labels. The resulting feature importance list quantifies the significance of each class label. Labels that have a greater influence on prediction accuracy receive higher importance scores, while less influential labels are assigned lower scores.

   Although this method appeared to be a reasonable approach, time limitations prevented its implementation and utilization in this study.

2. *Occurrence frequency*: this method involves using the base 2 logarithm of the occurrence frequency of categories in the BABEL dataset (Figure 4). The number of occurrences of action categories is considered a quantifiable measure for comparison, but to ensure comparability, the base 2 logarithm is applied to normalize the categories, as recommended by the creators of the BABEL dataset [37].

The first method seems to be a reasonable approach, but its usage in the literature was not documented for human motion prediction, indicating that it needed to be supported by a more formal study, which could not be explored because it was not the main focus of this thesis.

This lead to the second method to be implemented as the final approach, concluding the description of the algorithm presented in Section 3.4.

4 METRICS

4.1 METRICS USED IN LITERATURE

The following sections provide an overview of commonly utilized metrics in the field of human motion prediction, along with pertinent research studies that employed these metrics.

Furthermore, to gain a more in-depth understanding of the strengths and limitations of the employed metrics, it is beneficial to examine the different types of angle representation shown in Section Human Motion.

A significant portion of the existing literature in this field heavily relies on the H3.6M dataset [20] as well as an Euler angle-based metric for assessing performance.

In contrast, the newly introduced AMASS [28] dataset presents a substantial increase in sample size, offering approximately 14 times more data compared to the H3.6M dataset.

4.1.1 *SPL*

The primary reference for defining the metrics for our experiments has been the paper entitled "Structured Prediction Helps 3D Human Motion Modeling" [1] due to the following factors:

- *Focus of the paper and formalization*. Of the two main contributions of the paper, the first centered on defining meaningful measures of accuracy for pose predictions, for the task of human motion prediction.

- *Comparison with existing metrics*. The paper focused on the metric aspect, going to deep detail both formalizing the metrics math, comparing and evaluating state-of-the-art methods of quantifying pose prediction accuracy.

- *Usage in other works*. The metrics proposed by the SPL paper by Aksan et al. [1] were used in other notable and recent work [2].

- *Code availability and readability*. One aspect that should not be underestimated is how much the code provided by the authors is clean and understandable. This means that the work and time needed to implement their ideas into our project can be optimized, but most importantly, a more readable code is easier to implement, which leads to the crucial aspect of being less error-prone.

The principal contributions of the SPL paper by Aksan et al. [1] are:

1. defining a meaningful measure of accuracy for pose predictions, where lower errors correspond to favorable qualitative outcomes of human motion.
2. proposing a novel structured prediction layer that enhances the performance of existing models in this domain.

In the SPL paper, a review of the existing metrics and evaluation methods is presented, a few mentions are in the following list:

- In the work by Fragkiadaki et al. [14], their evaluations are conducted on the H3.6M dataset [20], employing a data representation based on joint angles represented by the exponential map, also known as the angle-axis representation. For evaluating the performance, the joint-wise Euclidean distance on the Euler angles is utilized as the evaluation metric.
- One of the most noteworthy performances achieved thus far on the H3.6M dataset is reported in the study conducted by Wang et al. [16] Their approach involves utilizing a sequence-to-sequence methodology, similar to previous works, but with the introduction of a geodesic loss (analogous to the joint angle difference metric 4.2.2) that holds greater significance in terms of accuracy assessment.

Proposed metrics

The metrics that were considered in the SPL [1] work are a conjunction of the most relevant and widely used metrics in the field of human motion prediction, and are listed below.

- **Euler angles error**: This widely adopted metric involves the specification of an Euler sequence (e.g. ZXY) and subsequently calculating the Euclidean distance between the predicted angles and the corresponding ground truth angles.
- **Joint angle difference**: Unlike being reliant on a specific angle parametrization, this metric quantifies the rotational angle required to align the predicted joint with the target joint. In the next sections (4.2), this will also be referenced as the "geodesic" metric.
- **Positional error**: By performing forward kinematics on the human kinematic model, this metric compares the positions of key points with the corresponding ground truth positions using the Euclidean distance (L2 norm).

- **PCK**: Originally introduced in the SPL paper, the Percentage of Correct Keypoints (PCK) metric computes the proportion of predicted joints that fall within a predetermined spherical threshold (ρ) around the target joint position.

  However, the selection of the threshold ρ often appears somehow arbitrary; hence it will not be utilized in the context of this thesis.

4.1.2 *History*

The positional metric involves comparing the predicted global positions of body keypoints to the ground truth values. This metric is important as it captures the cumulative effects of relative errors along the kinematic model. Unlike the Euler angle error metric, which focuses on relative joint angle errors, the positional metric provides a broader perspective on the overall accuracy.

This distinction arises because joints or body keypoints located at the edges of the kinematic tree have a relatively smaller impact on the global position compared to the main joints in the body. The errors in these main joints can significantly affect the global positions of the keypoints connected to them, resulting in a magnified effect on the overall prediction accuracy.

The paper "History Repeats Itself: Human Motion Prediction via Motion Attention" [30] presented one of the first models that could beat the Zero Velocity baseline model (more on the zero-velocity in the Section 4.3) and it specialized on the prediction of positions.

As a result, it introduces the joint position error and also employs the widely used Euler angles error for evaluating performance. Both these metrics will be used in our work, with the rigorous definitions found in the work by Aksan et al. [1]. This is also one of the reasons that the experimental results obtained by running their algorithms are used as a cross-reference in our work to validate our implementation of the zero-velocity model and the metrics implementation.

4.1.3 *Spatio-temporal Transformer*

The Euler error, joint angle difference and positional metrics were also employed in a recent research that involved transformer models[2]. Their implementation of the metrics followed the definitions found in the SPL paper [1].

The paper "A Spatio-temporal Transformer for 3D Human Motion Prediction" [2] proved to have a similar approach to metrics as the ones that will be implemented in this thesis.

The evaluation of the model's performance in this study involves the utilization of a mean angular error (MAE) on various metrics, including the L2 norm of Euler angles, positional metrics, and joint-angle difference. The MAE is defined as shown in Equation 13:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} |\hat{y}_n - y_n| \tag{13}$$

where $\hat{y}_n$ represents the predicted value and y_n represents the corresponding ground truth value. The MAE is calculated as the average absolute difference between the predicted and ground truth values across N instances. This metric provides a quantitative measure of the overall accuracy of the model's predictions.

4.2 METRICS DESCRIPTION

The metrics implemented in this work have been inspired by the SPL and Spatio-temporal Transformer papers [1, 2].

4.2.1 *Euler angles*

The Euler angle error is one of the most commonly utilized metrics in the field of human motion prediction. Despite its known limitations, it remains valuable for comparing the performance of our work and model with existing studies. While the Euler angle metric has certain weaknesses, it serves as a relevant benchmark for assessing the accuracy of joint angle predictions.

The Euler angle metric, denoted as $L_{\text{eul}}(t)$, is utilized to evaluate performance at time step t. It is computed as follows:

$$L_{\text{eul}}(t) = \frac{1}{|\mathcal{X}_{\text{test}}|}\sum_{\mathbf{x}_t \in \mathcal{X}_{\text{test}}} \sqrt{\sum_k \left(\boldsymbol{\alpha}_t^{(k)} - \hat{\boldsymbol{\alpha}}_t^{(k)}\right)^2} \tag{14}$$

The Euler angle metric evaluates the average Euclidean distance between the predicted and ground truth Euler angles across all samples in the test set. While this metric has certain weaknesses, it can be used with other metrics as a benchmark for assessing the accuracy of joint angle predictions.

4.2.2 *Joint-Angle Difference*

To mitigate potential errors associated with the Euler angle metric, SPL [1] proposed this alternative angle-based metric, which quantifies the angle of rotation required to align the predicted joint with the target joint. Unlike the Euler angle metric, the geodesic metric is independent of the specific parameterization of rotations and exhibits similarities to the geodesic loss. It is defined in Equation 15

$$L_{\text{angle}}(t) = \frac{1}{|\mathcal{X}_{\text{test}}|} \sum_{\mathbf{x}_t \in \mathcal{X}_{\text{test}}} \frac{1}{K} \sum_{k} \left\| \log\left(\tilde{\mathbf{R}}_t^{(k)}\right) \right\|_2 \tag{15}$$

Here, $\tilde{\mathbf{R}}t^{(k)}$ represents the rotation matrix of joint k at time t. It is important to note that unlike the Euler angle metric, which operates on local joint angles, the geodesic metric evaluates the loss on global joint angles by unwinding the kinematic chain before the computation of Langle.

In this thesis, we have also introduced the "local" joint angle difference metric to assess relative angle errors, similar to the Euler angle error approach. This addition is motivated by the nature of our Multipose model, which consists of N independent prediction models.

Each model generates predictions for one of the N joints in the simplified kinematic model of the human body. Thus, this metric provides an evaluation of how effectively each individual model predicts the joint angles. The local joint angle difference metric calculates the average performance across all models, offering insights into the overall predictive capabilities of the Multipose model.

4.2.3 *Positional*

To measure the accuracy of joint (or more precisely, body keypoint) positions, we employ the positional metric, denoted as $L_{\text{pos}}(t)$.

The positional error metric offers a more comprehensive perspective on the overall error by considering the influence of errors in the initial joints of the kinematic tree, such as the spine and shoulders. These initial joints have a greater impact on determining the final positions of the subsequent joints compared to joints like the wrists. Consequently, evaluating the positional error metric allows for a more holistic assessment of the error, considering the cumulative effect of errors across multiple joints rather than focusing solely on individual joint relative errors.

This metric involves performing forward kinematics on the predicted joint positions $\hat{\mathbf{p}}_t$ and the ground truth joint positions $\mathbf{p}_t$ at time t. This process yields 3D joint positions p_t and $\hat{p}_t$, respectively. The Euclidean distance between each corresponding joint position pair is then computed.

$$L_{\text{pos}}(t) = \frac{1}{|\mathcal{X}_{\text{test}}|} \sum_{\mathbf{x}_t \in \mathcal{X}_{\text{test}}} \frac{1}{K} \sum_{k} \left\| \mathbf{p}_t^{(k)} - \hat{\mathbf{p}}_t^{(k)} \right\|_2 \tag{16}$$

where $|\mathcal{X}_{\text{test}}|$ represents the total number of test samples, $\mathbf{x}_t$ denotes an individual sample at time t, K indicates the total number of joints, and $\mathbf{p}_t^{(k)}$ and $\hat{\mathbf{p}}_t^{(k)}$ refer to the ground truth and predicted joint positions, respectively. To ensure consistency, the lengths of the skeleton bones are normalized, with the right thigh-bone serving as a unit length reference as suggested in [1].

4.3 BASELINE MODEL: ZERO VELOCITY MODEL

To evaluate the prediction quality of our model, metrics are employed; however, these metrics alone may not provide a comprehensive assessment of the model's performance. Hence, it is crucial to compare the model's metrics with those of a suitable baseline. In our study, we have chosen the Zero-Velocity model as the baseline for comparison.

It is worth noting that other studies, such as Martinez et al. [32], have utilized different baseline models, such as the running average.

In numerous cases, the zero velocity model has proven to be challenging to surpass, particularly concerning Euler joint angle metrics, as highlighted in the work by Aksan et al. [1]. Thus, comparing our model's metrics against the zero velocity model provides valuable insights into the model's performance and its ability to outperform the baseline in terms of joint angle predictions.

The Zero-Velocity model: despite its simplicity, it exhibits competitive performance even when compared to some deep learning models. This model operates on a straightforward principle, making its performance against more complex models quite remarkable, highlighting the significance of comparing our model's performance against this baseline.

The Zero-Velocity model can be defined as a zero-order hold of the most recent observed frame. In practical terms, this means that every predicted frame will be an exact replica of the joint angles present in the last observed frame.

4.3.1 *Validation of metric and zero-velocity implementations*

To validate the accuracy of our implemented metrics and the Zero-Velocity model, we conducted testing using the results obtained from the STT paper [2] on the DIP dataset [19].

The Euler angle error, positional error, and global joint angle difference exhibited results within a 1% margin of the values reported in the Spatio-temporal Transformer paper [2]. This alignment with the reputable reference validates the reliability of our metrics and the effectiveness of the Zero-Velocity.

5 TESTING AND RESULTS

5.1 SETUP AND RESOURCES

This thesis made use of two primary computing resources. The first resource, referred to as the *C-square Lab* in Table 2, was employed for the development of the model and metrics. Notably, this computer was also utilized in the development of the MultiPose model by Matteo Cunico [12].

The Cluster DTG (see Table 2) computer was utilized for computationally intensive tasks involving dataset training and preprocessing. This computing resource played a crucial role due to the limitations of the C-square Lab computer, particularly when handling the preprocessed data with additional semantic information, which was significantly larger.

Moreover, leveraging multiple GPUs on the Cluster DTG computer enabled parallel training of the models, leading to a substantial reduction in training time.

Table 2: Computer specifications.

Computer	**C-square Lab**	**DTG Cluster**
GPU	NVIDIA GeForce RTX 2080 Ti	3x NVIDIA Tesla V100 S
GPU mem	11 GB	3x 32 GB
CPU	Intel Core i9-9900K	Intel Xeon CPU E5-2609 v3
RAM	32 GB	64 GB
Disk	250 GB	30 TB

5.2 HYPERPARAMETER OPTIMIZATION

Several model hyperparameters were considered during the experimentation, including semantic scene dimension, decoder, encoder hidden sizes, learning rate, weight decay, teacher forcing, number of training epochs, and batch size. Some key aspects are elaborated upon below:

- *Layer dimensions*: The hidden layer dimensions of the encoder and decoder play a crucial role in determining the complexity of the models. These dimensions significantly impact both training time and prediction accuracy.

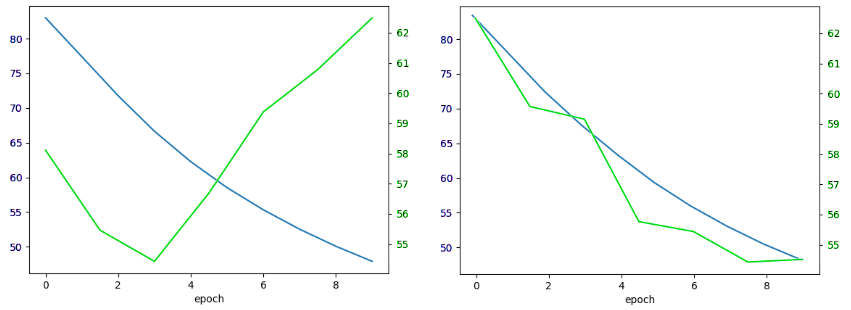
- *Framerate and sequence lengths*: The motion capture data has been downsampled to a fixed framerate of 25Hz, with an observation window of 0.8s and a desired predicted sequence length of 1.2s, so the model receives 20 frames as input and is trained with 30 frames as output, but could theoretically produce longer sequences thanks to the recursive nature of the decoders.

Number of training epochs

About the number of epochs and the occurrence of overfitting in our models, an observation is that each joint model exhibited a slightly different optimal point. Specifically, we noticed that some joint models tended to overfit early on (as in number of training epochs), as demonstrated in Figure 6a, which represents the training of the model dedicated to the global orientation joint of the body.

Conversely, Figure 6b illustrates that joints such as the spine follow a more meaningful (less randomic) movement, can lead to better training

The reason behind this phenomenon lies on the fact that some joints, like the global orientation of the body, can assume almost any value (along the vertical axis of rotation), making it challenging for the model to make accurate predictions and to train effectively.



(a) Evaluation on validation set during training (SMPL joint index 0, see Figure 2).

(b) Loss function trend during training epochs for one of the spine joints (SMPL joint index 6, see Figure 2).

Figure 6: Loss and evaluation plots to verify overfitting of the models.

5.3 MODELS COMPARISON: WITH AND WITHOUT SEMANTICS

All models in this study have been trained on the BMLrub dataset [44]. To ensure consistent evaluation, the dataset's mocap recordings were randomly divided into three distinct groups: 85% of the mocap data was allocated for training purposes, 10% for the validation set, and the remaining 5% constituted the test subset. The results presented in this section exclusively pertain to the test subset, which was employed for reporting the outcomes of our experiments.

Training and inference times

The trained models used in this study are presented below. It is important to note that the numerical values in the model's code-name (32 and 128) represent the hidden sizes of the encoders and decoders.

- *Small model* (*32 kin*): This model was trained to establish a baseline for assessing the capabilities of the system under capacity limitations (low number of parameters as compared to the original Multiverse model [24]). This model was created because the training initially took place on the C-square Lab computer, which has inherent constraints compared to the Cluster DTG (refer to Table 2). The model exclusively utilizes kinematic information and does not incorporate semantic data.

- *Small semantic model* (*32 kin+sem*): trained to serve as a direct comparison to the "*32 kin*" model. However, in this case, semantic information was integrated into the training (and inference) process.

- *Big model* (*128 kin*): This model was given a significantly larger number of parameters and focuses exclusively on kinematic information.

- *Big semantic model* (*128 kin+sem*): this serves as a comparison to the "*128 kin*" model, but with the inclusion of semantic information. It represents the final model and is expected to yield superior predictive performance when provided with semantic context as inputs.

Table 3 shows the training times for the different models.

The dataset preprocessing with semantics requires approximately six times more time compared to the non-semantic one. Specifically, for the training dataset alone (BMLrub [44]), the models without semantic information require around 10 minutes for preprocessing, while the inclusion of semantic information extends the preprocessing time to approximately one hour.

Table 3: Training times of the models. * : indicates that the model has been trained on the C-square Lab computer, the others were trained on the Cluster DTG, see table 2

	single model [minutes]	total [hours]
32 kin*	47	15,0
32 kin+sem	16	5,2
128 kin	34	10,7
128 kin+sem	38	12,0

Inference Times: Utilizing the available hardware, the small (32) and big (128) models require approximately 2.2s and 3s, respectively, to generate predictions for the desired 1.2s sequence. It is important to note that these inference times come from models that are not optimized for real-time performance. However, with further optimizations and dedicated hardware, these models hold potential for effective real-time motion prediction applications. For instance, instead of having independent models for each joint, a single model could work faster because it would have to load less data which is in common with all the joints (such as the semantic data).

5.3.1 *Metrics*

This section presents the comprehensive results obtained using the proposed metrics, including Euler angle error (Table 4), global and local joint angle difference (Tables 6 and 5), and positional error (Table 7). The metrics are computed in 200ms time increments and stop at 1s of prediction time.

The presented metrics were obtained from the inference of 279 mocap sequences extracted from the selected subset of AMASS dataset [44]. It is important to note that, to better represent realistic scenarios, not all mocap sequences possess semantic information pertaining to the corresponding actions they represent. As indicated in Table 1, approximately 60% of the mocap sequences in the dataset include semantic information. This reflects the inherent nature of real-world scenarios, where the actions performed by human agents may not align with the predefined semantic classes (macrocategories) and could be undefined.

Cell Coloring: The cells within the tables have been color-coded according to specific columns. A darker shade of red indicates a higher error, while a darker shade of green signifies a lower error, indicating a more accurate prediction. Cells that are white or lightly colored fall within the middle range of error for the respective column.

Table 4: Cumulative and windowed average Euler angle error comparison. Lower is better.

	Cumulative					Avg window (3 frames)				
Time [ms]:	**200**	**400**	**600**	**800**	**1000**	**200**	**400**	**600**	**800**	**1000**
Zero-Vel	4.14	10.67	18.05	25.24	31.90	0.94	1.37	1.49	1.42	1.33
32 kin	4.83	10.15	15.99	22.16	28.90	0.93	1.09	1.19	1.24	1.38
32 kin+sem	5.31	10.98	17.12	23.53	30.36	1.01	1.15	1.25	1.28	1.40
128 kin	4.64	10.02	15.91	22.19	28.86	0.90	1.10	1.20	1.26	1.37
128 kin+sem	4.44	9.60	15.33	21.49	28.06	0.86	1.06	1.17	1.24	1.35

The findings indicate that the model with the highest capacity, along with the inclusion of semantic information (*128 kin+sem*), emerges

Table 5: Cumulative and windowed average joint angle difference on local (relative) joint angles. Lower is better.

	Cumulative					Avg window (3 frames)				
Time [ms]:	**200**	**400**	**600**	**800**	**1000**	**200**	**400**	**600**	**800**	**1000**
Zero-Vel	0.75	1.94	3.31	4.65	5.89	0.17	0.25	0.28	0.26	0.25
32 kin	0.94	1.96	3.08	4.26	5.54	0.18	0.21	0.23	0.24	0.26
32 kin+sem	1.05	2.13	3.32	4.56	5.87	0.20	0.22	0.24	0.25	0.27
128 kin	0.90	1.90	3.01	4.20	5.46	0.17	0.21	0.23	0.24	0.26
128 kin+sem	0.87	1.83	2.92	4.09	5.33	0.17	0.20	0.22	0.24	0.25

Table 6: Cumulative and windowed average joint angle difference on global (absolute) joint angles. Lower is better.

	Cumulative					Avg window (3 frames)				
Time [ms]:	**200**	**400**	**600**	**800**	**1000**	**200**	**400**	**600**	**800**	**1000**
Zero-Vel	1.08	2.87	4.99	6.94	8.67	0.25	0.38	0.43	0.38	0.34
32 kin	1.34	2.79	4.38	6.02	7.76	0.26	0.30	0.32	0.33	0.36
32 kin+sem	1.57	3.21	4.99	6.77	8.60	0.30	0.33	0.36	0.35	0.37
128 kin	1.27	2.69	4.28	5.91	7.64	0.24	0.29	0.32	0.33	0.35
128 kin+sem	1.27	2.69	4.25	5.86	7.56	0.24	0.29	0.32	0.32	0.35

Table 7: Cumulative and windowed average positional error comparison. Lower is better.

	Cumulative					Avg window (3 frames)				
Time [ms]:	**200**	**400**	**600**	**800**	**1000**	**200**	**400**	**600**	**800**	**1000**
Zero-Vel	0.43	1.16	2.02	2.86	3.66	0.10	0.15	0.18	0.16	0.16
32 kin	0.49	1.07	1.73	2.45	3.25	0.10	0.12	0.13	0.14	0.16
32 kin+sem	0.58	1.22	1.95	2.72	3.54	0.11	0.13	0.15	0.15	0.17
128 kin	0.47	1.03	1.69	2.40	3.18	0.09	0.12	0.13	0.14	0.16
128 kin+sem	0.46	1.00	1.64	2.33	3.09	0.09	0.11	0.13	0.14	0.16

as the top-performing model, surpassing all others across nearly all metrics.

Conversely, the *32 kin+sem* model consistently demonstrates poorer performance, occasionally on par with or even worse than the zero-velocity model. On the other hand, the *32 kin* model falls in between the best and worst models, exhibiting moderate performance relative to the others.

A noteworthy comparison to is between the "global" and "local" metrics. Here, the term "global" refers to absolute metrics such as positional error and global joint angle difference, while the "local" metrics pertain to the Euler angle error and the local joint angle difference metric, which are computed on the relative joint angles.

In terms of the global metrics, it appears that the larger models generally exhibit an advantage, while this discrepancy is less pronounced in the local metrics. This observation can be attributed to the nature of the MultiPose model, which consists of multiple independent models. Each individual model predicts a joint angle without considering the context of the other joints.

It appears that the smaller models perform reasonably well in the relative metrics, but face challenges in the global metrics, where the MultiPose model inherently encounters greater difficulty due to the independent nature of the individual joint models.

First prediction frames

The resulting metrics tables, provide insightful observations when comparing cumulative and average window errors. For the first predicted frames (at time $200ms$), the zero-velocity model demonstrates relatively better performance in terms of cumulative metric compared to other models. However, upon examining the windowed average, which considers the three preceding frames, the zero-velocity model is no longer among the top performers.

This pattern is not unique to our motion prediction models; it is commonly observed in various models. The initial instant of prediction often suffers from the inherent discontinuity between the observed pose and the predicted pose. Consequently, it is natural that the zero-velocity model, which remains stationary, does not exhibit significant errors during the initial divergence of movements from the last observed pose.

Curse of Dimensionality

Across all metrics, the performance of the "small" model with semantics (*32 kin+sem*) consistently exhibits lower performance. This suggests that the introduction of semantic information may pose challenges for a model with a limited number of parameters. Notably, the model with equivalent capacity but without semantic information (32 kin) demonstrates comparatively better performance.

The phenomenon known as the "curse of dimensionality" becomes apparent when the addition of semantic information overwhelms a model with limited parameter capacity. The increased dimensionality introduced by semantic information may hinder the model's ability to effectively learn and generalize.

5.4 MULTIFUTURE PREDICTIONS

The MultiPose model is designed to generate multi-future predictions by leveraging the classifier (2.2.2.2) and employing diverse beam search (Diverse beam search). While individual joint predictions, produced by separate models for each kinematic model joint, exhibit some diversity due to the application of diverse beam search, the final predictions do not.

Note that the final "body" prediction is obtained through a beam search, exploring the single joint predicted trajectories and generating the full-body pose.

Upon evaluating the models, it became apparent that the generated future trajectories exhibit minimal deviations from one another, indicating a lack of qualitative differences between them.

Increasing the search diversity strength parameter (Diversity Strength) does enhance the diversity of individual joint trajectories (although at the expense of decreased prediction accuracy). However, the final body predictions, still exhibit considerable similarity to one another. This observation may suggest the suboptimal nature of the dataset employed for training the models may contribute to this behavior in the multifuture prediction task.

Understanding whether the dataset allows for sequences of movements that share an initial portion but diverge into different movements is not a trivial task. To address our specific needs, it would be beneficial to have instances with identical initial contexts but different outcomes, which would be highly valuable for robotics. Upon examining the utilized dataset [28], it makes sense that there is a limited multifuture diversity, since it is principally employed for action recognition tasks or single future predictions.

CONCLUSIONS

This study showed how semantics was integrated into the MultiPose model, which is a model for predicting human motion. Additionally, it provided a comprehensive review of the relevant literature on the methodological aspects of both existing motion prediction models and metrics to evaluate those models.

The primary objective of enhancing the predictive capabilities of the MultiPose model was achieved by incorporating semantics into the model's framework. This involved utilizing the BABEL dataset, which provided semantic labels for human motion sequences, in conjunction with the kinetic information provided by the AMASS dataset.

Through detailed evaluation, it was found that the integration of semantic information led to improvements in the prediction performance. Specifically, the more extensive model exhibited significant improvements across the most used metrics, surpassing the performance of the models without semantic context.

The MultiPose model, designed with the highest number of parameters (controlled in major part by the sizes of the encoder and decoders hidden states), was tested alongside a model with the same parameter capacity but trained with only kinematic data. We also provide the metrics of the zero-velocity model as a common reference baseline, which has been validated with other works to ensure the validity of our results.

Despite the historical difficulty of outperforming the zero-velocity baseline even for deep learning models (with the most common metrics such as Euler error), the experiments revealed that the model with semantic information achieved lower average positional errors than the zero-velocity baseline within the first $800ms$ of prediction. Furthermore, at $1s$ of prediction, the error was on par with the baseline. This indicates that the model does not diverge worse than the baseline, which is a positive indication for the possibility of long-term prediction.

We observed that the multifuture aspect of the model could be improved with the employment of a dataset with enhanced motion diversity, as in a more diverse "branching" of actions given similar initial contexts.

Another aspect that was found with extensive testing is that models which employ semantic information should be designed with the right parameter capacity (or model complexity) to be able to effectively enhance the prediction accuracy rather than models which only employed kinematic information.

These conclusions open a path for future work, since we demonstrated that the addition of semantic information improves prediction performance. This means that many existing architectures, which only employ kinematic motion information, could be improved by exploring the semantic aspect.

While this thesis focused on the semantics related to the action categorization, other types of semantics can improve the predictions, as shown in other works which employed attention mechanisms to identify semantic correlations between body parts. Other types of semantics can derive from the environmental data, such as object handling, physical obstacles or the presence of multiple actors in the scene.

Ultimately, the goal of these research directions is to further improve human-robot collaboration, by enhancing safety and efficiency in industrial settings and this thesis proved to be a step in the right direction for enhancing the performance of existing motion prediction models.

BIBLIOGRAPHY

[1] Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. Structured prediction helps 3d human motion modelling. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7143–7152, 2019.

[2] Emre Aksan, Peng Cao, Manuel Kaufmann, and Otmar Hilliges. A spatio-temporal transformer for 3d human motion prediction. *2021 International Conference on 3D Vision (3DV)*, pages 565–574, 2020.

[3] Adel Ammar, Anis Koubaa, Wadii Boulila, Bilel Benjdira, and Yasser Alhabashi. A multi-stage deep-learning-based vehicle and license plate recognition system with real-time edge inference. *Sensors*, 23(4), 2023. URL https://www.mdpi.com/1424-8220/23/4/2120.

[4] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *CoRR*, abs/1506.03099, 2015. URL http://arxiv.org/abs/1506.03099.

[5] Ananth Reddy Bhimireddy, Priyanshu Sinha, Bolu Oluwalade, Judy Wawira Gichoya, and Saptarshi Purkayastha. Blood glucose level prediction as time-series modeling using sequence-to-sequence neural networks. In *KDH@ECAI*, 2020.

[6] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields, 2019.

[7] Carnegie Mellon University. CMU MoCap Dataset, unknown. URL http://mocap.cs.cmu.edu.

[8] Cristian Sminchisescu Catalin Ionescu, Fuxin Li. Latent structured models for human pose estimation. In *International Conference on Computer Vision*, 2011.

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, 2017.

[10] Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Z. Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Katya Gonina, Navdeep Jaitly, Bo Li,

Jan Chorowski, and Michiel Bacchiani. State-of-the-art speech recognition with sequence-to-sequence models. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778, 2017.

[11] Enrique Coronado, Takuya Kiyokawa, Gustavo A. Garcia Ricardez, Ixchel G. Ramirez-Alpizar, Gentiane Venture, and Natsuki Yamanobe. Evaluating quality in human-robot interaction: A systematic search and classification of performance and human-centered factors, measures and metrics towards an industry 5.0. *Journal of Manufacturing Systems*, 63:392–410, 2022. URL https://www.sciencedirect.com/science/article/pii/S0278612522000577.

[12] Matteo Cunico. Human motion anticipation through 3d structured multi-future trajectory prediction. *University of Padua, Dipartimento di Tecnica e Gestione dei Sistemi Industriali - DTG*, December 2021. URL https://hdl.handle.net/20.500.12608/41371.

[13] Scott L. Delp, F. Clayton Anderson, Allison S. Arnold, Peter Loan, Ayman Habib, Chand T. John, Eran Guendelman, and Darryl G. Thelen. Opensim: Open-source software to create and analyze dynamic simulations of movement. *IEEE Transactions on Biomedical Engineering*, 54:1940–1950, 2007.

[14] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics, 2015.

[15] Saeed Ghorbani, Kimia Mahdaviani, Anne Thaler, Konrad Kording, Douglas James Cook, Gunnar Blohm, and Nikolaus F. Troje. MoVi: A Large Multipurpose Motion and Video Dataset. *Borealis*, V5, 2020. doi: 10.5683/SP2/JRHDRN. URL https://doi.org/10.5683/SP2/JRHDRN.

[16] Liang-Yan Gui, Yu-Xiong Wang, Xiaodan Liang, and Jose M. F. Moura. Adversarial geometry-aware human motion prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[17] Kengo Hotate, Masahiro Kaneko, and Mamoru Komachi. Generating diverse corrections with local beam search for grammatical error correction. In *International Conference on Computational Linguistics*, 2020.

[18] Pengpeng Hu, Edmond Ho, and Adrian Munteanu. 3dbodynet: Fast reconstruction of 3d animatable human body shape from a single commodity depth camera. *IEEE Transactions on Multimedia*, PP:1–1, 04 2021. doi: 10.1109/TMM.2021.3076340.

[19] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J. Black, Otmar Hilliges, and Gerard Pons-Moll. Deep inertial poser learning to reconstruct human pose from sparseinertial measurements in real time. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 37(6):185:1–185:15, November 2018.

[20] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014.

[21] Philipp Kratzer, Marc Toussaint, and Jim Mainprice. Prediction of human full-body movements with motion optimization and recurrent neural networks. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1792–1798, 2019.

[22] Alex Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks, 2016.

[23] Qin Li, Yong Wang, and Fanbing Lv. Semantic correlation attention-based multiorder multiscale feature fusion network for human motion prediction. *IEEE transactions on cybernetics*, PP, 2022.

[24] Junwei Liang, Lu Jiang, Kevin P. Murphy, Ting Yu, and Alexander G. Hauptmann. The garden of forking paths: Towards multi-future trajectory prediction. *CoRR*, abs/1912.06445, 2019. URL `http://arxiv.org/abs/1912.06445`.

[25] Hongyi Liu and Lihui Wang. Human motion prediction for human-robot collaboration. *Journal of Manufacturing Systems*, 44: 287–294, 2017. URL `https://www.sciencedirect.com/science/article/pii/S0278612517300481`. Special Issue on Latest advancements in manufacturing systems at NAMRC 45.

[26] Shaohua Liu, Shijun Dai, Jingkai Sun, Tianlu Mao, Junsuo Zhao, and Heng Zhang. Multicomponent spatial-temporal graph attention convolution networks for traffic prediction with spatially sparse data. *Computational Intelligence and Neuroscience*, 2021, 2021.

[27] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6): 248:1–248:16, October 2015.

[28] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *2019 IEEE/CVF International Conference*

on Computer Vision (ICCV), pages 5441–5450, October 2019. doi: 10.1109/ICCV.2019.00554.

[29] C. Mandery, Ö. Terlemez, M. Do, N. Vahrenkamp, and T. Asfour. The KIT whole-body human motion database. In *2015 International Conference on Advanced Robotics (ICAR)*, pages 329–336, July 2015. doi: 10.1109/ICAR.2015.7251476.

[30] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. History repeats itself: Human motion prediction via motion attention. In *European Conference on Computer Vision*, 2020.

[31] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. Generating smooth pose sequences for diverse human motion prediction. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13289–13298, 2021.

[32] Julieta Martinez, Michael Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *ArXiv*, 07 2017. doi: 10.1109/CVPR.2017.497.

[33] Graham Neubig. Neural machine translation and sequence-to-sequence models: A tutorial. *ArXiv*, abs/1703.01619, 2017.

[34] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019.

[35] Dario Pavllo, David Grangier, and Michael Auli. Quaternet: A quaternion-based recurrent model for human motion. *ArXiv*, abs/1805.06485, 2018.

[36] Nicola Pedrocchi, Federico Vicentini, Matteo Malosio, and Lorenzo Molinari Tosatti. Safe human-robot cooperation in an industrial environment. *International Journal of Advanced Robotic Systems*, 10:1–13, 01 2012. doi: 10.5772/53939.

[37] Abhinanda R. Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J. Black. BABEL: Bodies, action and behavior with english labels. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 722–731, June 2021.

[38] Md. Moshiur Rahman, Shajeeb Chakma, Dewan Mamun Raza, Sadia Akter, and Abdus Sattar. Real-time object detection using machine learning. *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–5, 2021.

[39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[40] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195, 2016.

[41] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), November 2017.

[42] Paul Schydlo, Mirko Raković, Lorenzo Jamone, and José Santos-Victor. Anticipation in human-robot cooperation: A recurrent neural network approach for multiple action sequences prediction. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–6, 2018.

[43] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *ArXiv*, abs/1406.2199, 2014.

[44] Nikolaus F. Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of Vision*, 2(5):2–2, September 2002. doi: 10.1167/2.5.2.

[45] Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *CoRR*, abs/1610.02424, 2016. URL http://arxiv.org/abs/1610.02424.

[46] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *European Conference on Computer Vision (ECCV)*, sep 2018.

[47] Ge Wu, Sorin Siegler, Paul Allard, Christopher Kirtley, Alberto Leardini, Dieter Rosenbaum, Mike Whittle, Darryl D. D'Lima, Luca Cristofolini, Hartmut Witte, Oskar Schmid, and Ian A. F. Stokes. Isb recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion–part i: ankle, hip, and spine. international society of biomechanics. *Journal of biomechanics*, 35 4:543–8, 2002.

[48] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI Conference on Artificial Intelligence*, 2018.

[49] Mohsen Zand, Ali Etemad, and Michael A. Greenspan. Flow-based autoregressive structured prediction of human motion. *ArXiv*, abs/2104.04391, 2021.