HAL open science

Motion2Language, Unsupervised Learning of Synchronized Semantic Motion Segmentation

Karim Radouane, Andon Tchechmedjiev, Julien Lagarde, Sylvie Ranwez

To cite this version:

Karim Radouane, Andon Tchechmedjiev, Julien Lagarde, Sylvie Ranwez. Motion2Language, Unsupervised Learning of Synchronized Semantic Motion Segmentation. Neural Computing and Applications, 2024, 36, p. 4401-4420. 10.1007/s00521-023-09227-z . hal-04251173

HAL Id: hal-04251173

https://imt-mines-ales.hal.science/hal-04251173

Submitted on 20 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MOTION2LANGUAGE, UNSUPERVISED LEARNING OF SYNCHRONIZED SEMANTIC MOTION SEGMENTATION

Karim Radouane[1], **Andon Tchechmedjiev**[1], **Sylvie Ranwez**[1], **Julien Lagarde**[2]
1. EuroMov Digital Health in Motion, Université de Montpellier, IMT Mines Alès, Alès, France
2. EuroMov Digital Health in Motion, Université de Montpellier, IMT Mines Alès, Montpellier, France

ABSTRACT

In this paper, we investigate building a sequence to sequence architecture for motion to language translation and synchronization. The aim is to translate motion capture inputs into English natural-language descriptions, such that the descriptions are generated synchronously with the actions performed, enabling semantic segmentation as a byproduct, but without requiring synchronized training data. We propose a new recurrent formulation of local attention that is suited for synchronous/live text generation, as well as an improved motion encoder architecture better suited to smaller data and for synchronous generation. We evaluate both contributions in individual experiments, using the standard BLEU4 metric, as well as a simple semantic equivalence measure, on the KIT motion language dataset. In a follow-up experiment, we assess the quality of the synchronization of generated text in our proposed approaches through multiple evaluation metrics. We find that both contributions to the attention mechanism and the encoder architecture additively improve the quality of generated text (BLEU and semantic equivalence), but also of synchronization. Our code will be made available at `https://github.com/rd20karim/M2T-Segmentation/tree/main`.

Keywords Unsupervised learning, Semantic segmentation, Synchronized transcription, GRU, local recurrent attention.

1 Introduction

Motion-Language processing constitutes an emerging field within computer vision, with evident links to cognitive science, computational linguistics and knowledge engineering, robotics, social computing. The current prism of study of the interplay between motion and language is mainly focused on preliminary exploratory investigations pertaining to the general association between close descriptions of the movements and its parameters (mainly motion capture). The task of Motion to Language mapping or translation consists of mapping sensor-motion parameters (e.g. MoCap) to natural language descriptions. Currently, few datasets include direct motion-to-description correspondences, enabling the use of supervised machine learning, mainly the KIT Motion-Language Dataset (KIT-ML) [Mandery et al., 2016] and HumanML3D Guo et al. [2022a]. In this paper, we take interest in motion-to-language mapping, exploring different designs of seq2seq architectures. To our knowledge, ours is the first system for unsupervised learning of motion-to-language segmentation and synchronization. We show why classical seq2seq architectures are unable to learn synchronous correspondences and propose iterative evolutions of the architecture for unsupervised learning of semantic motion segmentation based attention weights. We evaluate the models using the BLEU@4 score, which we extend with a semantic coherence metric. Finally, to measure the performance of segmentation, we propose different methods to compute the evaluation metrics for the synchronicity of semantic segmentation. We manually annotate the timings of motion segments of a subset of the KIT-ML dataset (a portion of our test set).

Highlight of main contribution

1. *Quantitative and visual illustration of the limitations of recurrent encoder on alignment.*
2. *Introduction of recurrent local attention for synchronous generation of text along the motion.*
3. *Inference of motion segmentation using only attention weights.*

4. *Proposition of a method to evaluate segmentation results and synchronization performance.*
5. *Manual annotation of the timings of motion segments on a subset of the test set, and an additional examination of data set descriptions after automated orthographic and grammatical corrections.*

2 Related work

In this section we will cover three main areas of relevance to our direct contributions, namely motion-language datasets and systems (main focus of the paper) and human motion segmentation (the main use-case for our approach).

2.1 Motion-Language Datasets and Systems

There are two public datasets mapping human pose sequences to linguistic descriptions: the KIT Motion Language dataset (KIT-ML) Plappert et al. [2016] and the HumanML3D (HML3D) Guo et al. [2022a]. The former is build by crowdsourcing annotations on the KIT Whole-Body Human Motion Database, and the latter by expert annotation of a subset of the AMASS dataset. Guo et al. Guo et al. [2022a] also introduce an augmentation of KIT-ML using the same process as for HumanML3D.

Subset	Number	Train	Test	Val.
KIT-ML	motions	2375	291	262
	samples	5071	612	568
KIT-ML-aug	motions	4886	830	300
	samples	10408	1660	636
HML3D-aug	motions	22068	4160	1386
	samples	66734	12558	4186

Table 1: Training splits dans dataset statistics, for KIT-ML and Human ML3D Guo et al. [2022a]. (-aug: after augmentation)

The first deep learning approach on the KIT-ML dataset was introduced by Plappert et al. Plappert et al. [2017]. They created a system capable of generating movement from textual descriptions and generating text that describes movement using a bidirectional LSTM encoder-decoder architecture. Later systems mainly focused on language to motion generation Lin et al. [2018], Ghosh et al. [2021], Petrovich et al. [2022], but motion to language generation has seen a resurgence with the introduction of HumanML3D.

Gotsu et al. Goutsu and Inamura [2021] explore adversarial seq2seq architectures for motion-to-language generation, achieving SOTA on KIT-ML. Guo et al. Guo et al. [2022b] take a bidirectional motion quantization based transformer approach, and are the first to use HumanML3D for motion-to-language generation. Although they achieve SOTA for motion generation on HumanML3D (the first system evaluated on the dataset). The transformer based method have not led to higher result on reverse direction (motion2text) specifically on KIT-ML (BLEU4 18.4%).

2.2 Human motion segmentation

There is no unified and consensual definition of motion segmentation, here we will focus specifically on approaches on skeleton data, which is what interests us. Lin and Kulič Lin and Kulic [2014] identify and segment movement repetitions using velocity features and stochastic modeling to select motion segment candidates, then, they apply a Hidden Markov Model (HMM) to select the final segment. Kilič et al. In the same vein, Dana et al. Kulić et al. [2009] learn to extract human motion primitives (*Lower arm raise*, *Bow Up/Down*,...) based on HMMs from a continuous time-series data converted to joint angles. Mei et al. Mei et al. [2021] consider segmentation from the angle of the unsupervised segmentation of human limb motion sequences using an autoregressive moving-average model (ARMA) to fit each limb bone angle, and defining segment points as the position where ARMA predictions no longer satisfy goodness of fit. A technique based on a collaborative representation of 3D skeleton sequences was proposed by Li et al. Li et al. [2018], where Aligned Cluster Analysis (ACA) Zhou et al. [2008] and its Hierarchical (HACA) variant Zhou et al. [2013] are used to identify sequences. More recent works Ma et al. [2021], Filtjens et al. [2022] rely on spatial temporal graph convolutions networks. But to our knowledge, no previous work was done in the aim to perform motion segmentation by synchronizing text generation to human skeleton motion.

3 Methods

Motion to language generation consist in generating a natural language description y of a motion sequence x, which can be seen as a sequence to sequence (seq2seq) mapping task, and more specifically a multi-modal Neural Machine Translation (NMT) system. In this paper, we propose a model that goes beyond simple mapping, by adapting NMT architectures to generate text aligned with motion. We explore both:

- **Architectural variations** The first variant use the GRU as motion encoder, the second variant use the Bidirectional GRU. For the third variant we propose to use a frame-level feature extractor based on the MLP that forgoes source summarization. Fig.1 gives an overview of the considered architecture variations.
- **Attention mechanism** We consider soft and local attention baselines, and our proposed recurrent local attention tailored for unsupervised synchronous text generation.

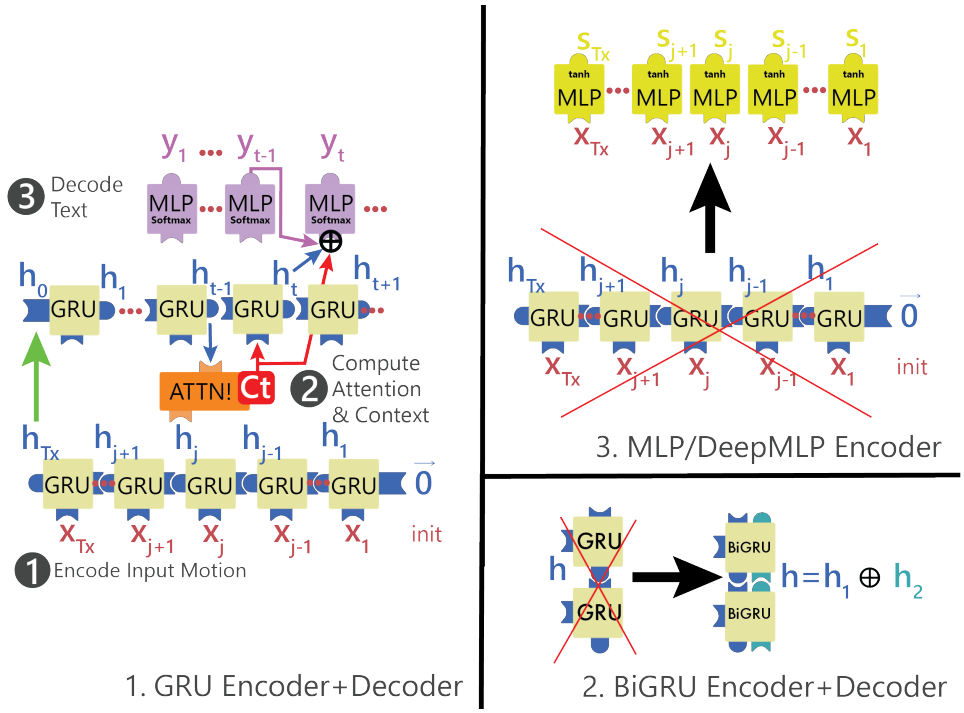


Figure 1: Considered Architectures.

As illustrate in the Fig. 1 We experiment three different motion encoders GRU, Bi-GRU and MLP with a GRU decoder. The x_j vector is the raw pose data of human skeleton at frame j. The output of the encoder frame j and the output of the decoder GRU cell h_{t-1} are used to compute attention weights and in turn the context vector C_t at token t. The previous word embedding y_{t-1}, the hidden state of the GRU h_t and the context vector C_t are concatenated and given as input to the final MLP/softmax layer. This layer predicts a probability distribution over the vocabulary of the language model. At each time step, the word with maximum probability, y_t is generated using a greedy decoding approach. This token is fed into the GRU for the next time step, and the process is repeated until the end token $<$ eos $>$ is reached. The Fig.2 specify the phases order of language prediction with respect to attention weights α_{it} used for segmentation.

In the remainder of the section, we discuss the classical NMT architectures and the role of soft and local attention in the computation of the sequence alignment (Section 3.1.1), and outline the main limitations of soft and local attention if we want to achieve synchronous generation (Section 3.1.2). We then introduce local recurrent attention, a novel formulation that allows synchronicity during generation (Section 3.2). Finally, we present an evolution of the encoder architecture better suited for small data and improving on synchronous generation (Section 3.3).

3.1 Baseline NMT Architecture and limitations

We first briefly present the baselines and their formulation, followed by a detailed analysis and discussion of our proposed evolution of the architecture to enable better synchronous generation. In this work, for all baseline variants, we

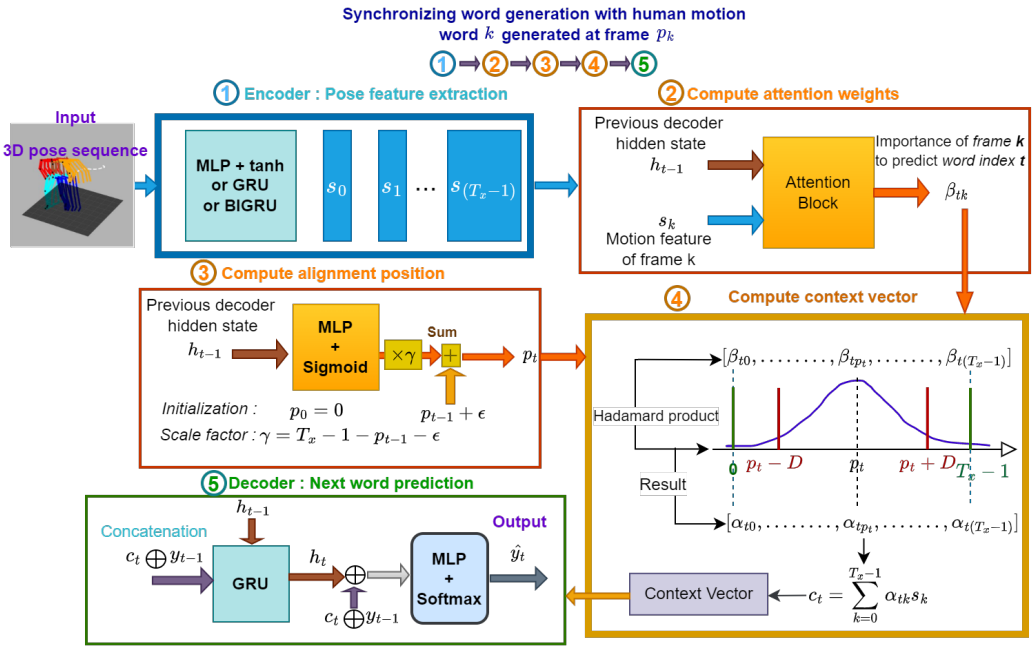


Figure 2: Phases of motion to language generation.

use a canonical NMT architecture Sutskever et al. [2014] adapted to receive poses as input instead of the source language tokens. The model is a Seq2seq Encoder/Decoder network, where the Encoder and Decoder are based on a Gated Recurrent Unit (GRU). The architecture variants are trained to model the probability $p(y \mid x) = \prod_{t=1}^{|y|} p(y_t \mid y_{<t}, x)$ through the minimization of the cross-entropy loss.

3.1.1 Classical global formulations of attention

Soft attention formulation The first type of attention considered is soft attention, the cannonical formulation for NMT presented by Bahdanau et al. Bahdanau et al. [2015]. The attention coefficients α_{ij} (Equation 1) are computed over the energy of possible alignments (Equation 2).

$$\alpha_{ij} = softmax([e_{i1}, \cdots, e_{iT_x}]) \tag{1}$$

$$e_{ij} = v_a^\top \tanh\left(W_a h_{t-1} + U_a s_j\right) \tag{2}$$

Here, T_x is the length of the motion sequence, $h_{t-1} \in \mathbb{R}^{n'}$ the decoder hidden state at word $t-1$, s_j the encoder output for motion frame j, while $v_a \in \mathbb{R}^{n' \times 1}$, $W_a \in \mathbb{R}^{n' \times n}$ and $U_a \in \mathbb{R}^{n' \times m}$ are learnable parameters, with $m = 1$ for the unidirectional GRU and MLP-based encoder, and $m = 2$ for the bidirectional GRU.

Local attention formulation The soft attention also allows the decoder to have access to information about past and future frames, and doesn't incentive the model to learn a precise localization the motions that corresponds to a particular set of words. The model learns to predict the moment when maximal information is available (at the very end). One solution can be to limit the information visible to the decoder through local attention Luong et al. [2015], applying a window of weights that only retains a local interval around the current frame. Luang et al. Luong et al. [2015] propose to learn an aligned position p_t for each decoding step t. A window centered around the source position p_t is used to compute the context vector c_t. The sum is directly computed inside the window $[\![p_t - D, p_t + D]\!]$ [1], where D is a hyperparameter. For the computation of p_t, we use h_{t-1} instead of h_t as formulated by Bahdanau et al. Bahdanau et al. [2015] as show in Equation 3.

$$p_t = T_x . \sigma({v_p}^T \tanh(W_p h_{t-1})) \tag{3}$$

[1] $[\![i, j]\!]$ denotes set of integers between i and j both included. We will use the Bourbaki convention throughout the paper.

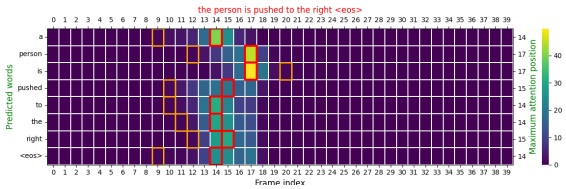
Where, $\sigma(.)$ denotes the sigmoid function, and where $W_p \in \mathbb{R}^{n' \times n'}$ and $v_p \in \mathbb{R}^{n' \times 1}$ are learnable parameters. In this work we consider a gaussian windowing function as it leads to smoother generation (see Equation 4).

$$a_{ij} = \alpha_{ij} \cdot \exp(-\frac{(j - p_t)^2}{2r^2}) \tag{4}$$

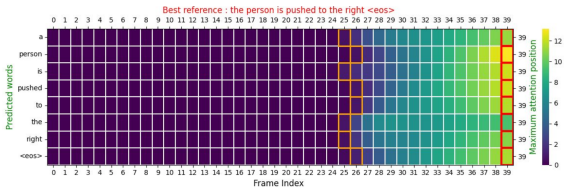
In all the following preliminary experiments we note by *Soft.att*, *Local.att*, *Local.rec.att* respectively the soft attention, local attention and local recurrent attention. The BLEU refers to the default BLEU@4 score. In all attention map Figures, the red rectangles represent the location of maximum attention for each word along the frame axis. The orange rectangles delimit the range of above zero activation.

3.1.2 Limitations of local and soft attention

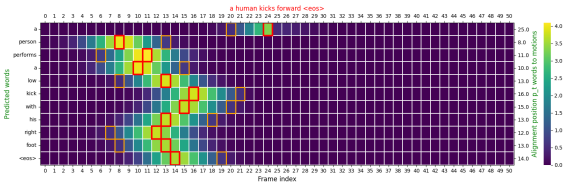
In order to observe the behavior and limitations of these classical formulations of attention and draw meaningful avenues of improvement, we perform a preliminary experiment, whereby we train the baseline GRU architecture (GRUenc + GRUdec), both unidirectional and bidirectional, on cartesian coordinate features, following the procedure described in Section 4, and plot the attention maps on a few representative motion samples.



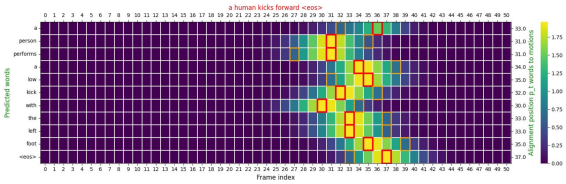
(a) BiGRU-cartesian[Soft.att] : Pushing action in the range $[\![11, 20]\!]$.



(b) GRU-cartesian[Soft.att] : Pushing action in the range $[\![11, 20]\!]$.



(c) BiGRU-cartesian[Local.att Mask False] : Kicking action in the range $[\![12, 24]\!]$.



(d) GRU-cartesian[Loc.att] : Kicking action in the range $[\![12, 24]\!]$.

Soft attention For BiGRU-Cartesian as we see attention in the Fig.3a we have a disorder in the position of maximum attention, which doesn't allow a synchronized generation of words. At this stage, the attention gives only correct information about the range of motion $[\![12, 21]\!]$ and action description. For the same sample in the case of unidirectional GRU, the maximum attention (figure 3b) was localized at the end frames. Observing the attention maps of the test set, observe that in general the maximal attention distribution was localized around the frame marking the end of the movement when the action is completed.

Local attention For BiGRU-cartesian, applying the mask results in the alignment positions being equal to the length of the motion. On the contrary, when the mask wasn't applied, as illustrated by the Fig.3c, surprisingly the alignment positions were well distributed in the range $[\![5, 24]\!]$, which corresponded to the interval of motion execution, but fell short of being well synchronized: for the word describing the action *"kick"* attention weights were correctly distributed in the range $[\![13, 19]\!]$ around the maximum attention frame 16 (center time of kick motion). For the same sample, in the case of the unidirectional GRU, the attention weights have a similar distribution, but in the range $[\![25, 41]\!]$, after the end of the action. In most cases, for the unidirectional GRU, the attention positions were distributed in the correct zones along the time/frame axis but not well synchronized.

Constraint on alignment position for synchronization

In order to obtain synchronous generation using only, $\boldsymbol{p_t}$ we derive the idea of calculating this position under the constraint that $\boldsymbol{p_{t-1}} \leq \boldsymbol{p_t}$. Although this constraint is language dependent and not universally true at the word level, it does hold up at the phrase level. For example, the words $\{"the", "a", "person"\}$ are not related to the monotony of frame generation, but for action words like ("walk", "jump"), the succession 'walk' then 'jump' happens successively in time, as results the word describing these actions will also appear successively in the human description. The description is made step by step as the motion is generated. The positions p_t will be more interesting for words that describe

actions. The other intermediate positions, for language connection, will serve as a relative position from which the next action is localized. When we have simultaneity of two actions described respectively by word w_i and w_j, ideally, the model will learn to predict approximately same position alignment $\boldsymbol{p_i} \approx \boldsymbol{p_j}$. Meaning that the position p_t will remain approximately constant in the range $[\![i, j]\!]$. On the side of language generation, the words connecting the two action words will be independent of motion. *The main goal is to associate every set of words in the sentence describing one action to the relevant set of frames based on p_t and attention weights.*

3.2 Introducing local recurrent attention

In local attention, the position p_t depends only on the previous hidden state h_{t-1}, which is problematic in our case, we observe that the network simply learns to almost always compute p_t at the start, middle, or at the end of the motion. The BiGRU is biased more towards the start and middle of the motion in Fig.3c (human movement end at the frame 27), while a unidirectional GRU favoured the positions after the end of the human movement in Fig.3d (same sample). Intuitively, the network gleans on compressed information about the entirely of the sequence and issues a prediction when the amount of information is maximal on average, thus negatively impacting its ability to perform synchronous predictions. Indeed, the network chooses the path of least effort without searching along the source length for finding the most relevant motion interval to output for a given phrase.

3.2.1 Formal definition

To overcome this limitation, we want the model to associate a subset of successive frames to the most relevant word(s) for an exact alignment and synchronization of segments of human motion with the generated text, and to this effect we propose a new formulation to compute the position p_t in Equation (5)[2], enforcing the important constraint that $\boldsymbol{p_{t-1}} \leq \boldsymbol{p_t}$ as explained in paragraph 3.1.2.

$$p_t = p_{t-1} + \epsilon + (T_x - 1 - p_{t-1}).\sigma(v_p^t \cdot W_p h_{t-1}) \tag{5}$$

The scaling factor $\boldsymbol{(T_x - 1 - p_{t-1})}$ maintains the position p_t within the range of the motion frames $[\![0, T_x - 1]\!]$. The position p_t indicate where to look in the source frames at a time t. The shift position $\boldsymbol{p_t - p_{t-1}}$ depends on the previous hidden state of the decoder h_{t-1}.

The learned motion segment S_t is delimited from the global sequence through masking [3] as defined by Equation (6). Then, using the recurrence relation in Equation (5) and the formulation of S_t in Equation (6), we can derive intersection segment boundaries as concluded in Equation (7). As mentioned previously, we use the Bourbaki notation for intervals, where $[\![i, j[\![$ refers to the set of integers between i and j (i included, j excluded).

$$S_t = [\![p_t - D, p_t + D[\![\cap [\![0, T_x[\![= [\![\max(0, p_t - D), \min(T_x, p_t + D)[\![\tag{6}$$

$$S_t \cap S_{t-1} = [\![p_{t-1} - D + \epsilon + (T_x - 1 - p_{t-1}).\sigma(v_p^t \cdot W_p h_{t-1}), \min(T_x, p_{t-1} + D)[\![\tag{7}$$

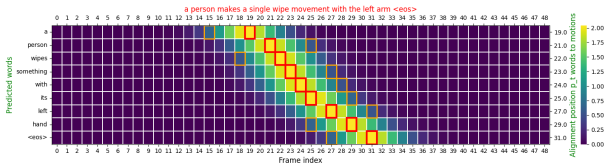
We define $\boldsymbol{\epsilon} \in \mathbb{N}$ as a parameter that can be used to force a minimum shifting between two successive positions and also reduce overlapping between successive local attention windows, by setting $\boldsymbol{\epsilon = (1 - \alpha).2D}$, where $\alpha \in [0, 1]$ is an explicit parameter to control overlapping proportion between successive motion segments $S_t \cap S_{t-1}$.

Using Equation (7) we have $\alpha = 0 \Rightarrow S_t \cap S_{t-1} = \emptyset$. This can be convenient to ensure proper word ordering and fast training in motions with clearly separate motion segments. Of course, in more complex cases such as simultaneous events, this mechanism is likely insufficient on its own. In addition, as a result of the recurrence effect, $\boldsymbol{p_t}$ is mechanically incremented by $\boldsymbol{\epsilon}$, meaning that $\boldsymbol{p_t \geq \epsilon \times t}$. This can lead the network to miss critical time intervals and to quickly go towards the final positions, particularly in very short and quick motions. To allow generating words describing simultaneous actions at the same time we set $\epsilon = 0$.

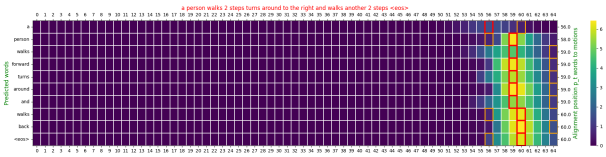
Also note that Equation (5) implicitly assumes that the motion and language description are *monotonically* related, meaning that the description of successive phases of a movement, will be in chronological order of the performance of the motion. For parameter D, we attempted to make it learnable through various formulations, but the values ended up being too high to allow for a correct segmentation in most cases, even when using a boundary function. However, when selecting its value as a hyperparameter we found a constant value for D leading to consistently better results.

[2] "." is the scalar product

[3] equivalent to applying a truncated Gaussian kernel



(a) GRU-cartesian[Local.rec.att Mask False]: Wiping motion in the range $[\![11, 32]\!]$.



(b) GRU-cartesian[Local.rec.att Mask False]: walk forward in the range $[\![15, 38]\!]$, turn at frame 40 , walk backward in $[\![42, 53]\!]$.

Context vector Noting by $s_x(j)$ the source hidden state at time $j \in [\![0, T_x - 1]\!]$ for a given motion x, the context vector is calculated for the two cases using Equation (8).

$$c_t = \begin{cases} \sum_{j=0}^{T_x-1} a_{tj} s_x(j) & \text{Mask is False} \\ \sum_{j \in T_{x_t}} a_{tj} s_x(j) & \text{Mask is True} \end{cases} \tag{8}$$

3.2.2 Applying recurrent attention

Again, in order to observe the behavior of our new formulation with regard to the classical formulations, we perform a second preliminary experiment, whereby we train the baseline GRU architecture (GRUenc + GRUdec), both unidirectional and bidirectional, following the procedure described in section 4, and plot the attention maps on a few representative motion samples.

Based on the attention maps for both cases Bi-GRU/GRU when applying the mask, the positions of maximum attention are always near the end of the motion sequence. On the contrary, when not applying the mask for the BiGRU, the attention weights were distributed progressively near the end of frames. However, for the unidirectional GRU we begin observing a progressive alignment that start at the end of the motion for some samples like in Fig.4a, the synchronization was more correct. But in all cases when we have more than one motion or action, like in Fig.4b, we see no synchronization between motion and language generation. This will be further explained and addressed in section 3.3 as well as the limitations related to this synchronization, as highlighted in section in 3.1.2.

3.3 Rethinking feature extraction for synchronous text generation

Recurrent effect of GRU. In the inspection, the alignments resulting from the GRU-based motion encoders with various attention formulations in the preliminary experiments, lead to an imperfect synchronization. We hypothesize that this is due to the recurrence effect present in the design of the GRU on the side of the encoder, its hidden state s_t depends on all previous states, and we can write $s_t = \varphi(s_{t-1}, x_t) = \varphi(\cdots \varphi(s_{t-\overrightarrow{k}}, x_{t-\overrightarrow{k}}))$. We note by φ the classical GRU model Equation and by $\overrightarrow{k}$, $\overleftarrow{k}$ respectively the limit level of compression without important information loss for the forward and backward direction. Hence, in the unidirectional case, s_t already encodes all the necessary information about the $\overrightarrow{k}$ previous steps $\{s_{t-\overrightarrow{k}}, s_{t-\overrightarrow{k}+1}, \cdots, s_{t-1}\}$. Putting maximum attention on s_t is equivalent to *seeing* the $\overrightarrow{k}$ previous motion frames: an action that happens at around frame time $t - \overrightarrow{k}$ can hold relevant global compacted information pertinent for frame t. The following clearly explains the misaligned position and late detection of primitive motion, as was illustrated by Figures 3b and 4b. For the Bi-GRU case, the analysis is similar, except that s_t now encodes all the necessary information about the $\overrightarrow{k}$ previous steps and $\overleftarrow{k}$ next information, this explains the localization of attention weight distribution towards the middle of motion segments (See Fig.3a). Following this rationale, we have determined that a frame-level feature extractor could resolve the late detection problem. Frame-wise feature extraction will push the network to use all source sequence features separately. The adoption of this type of feature extraction, matches similar choices in other works pertaining to alignment tasks, such as sign language/movie subtitle transcription Bull et al. [2020, 2021]. Although, the choice isn't supported by quantitative evidence. The main difference in our approach is that we do not have a supervised timing annotation of movement phases/primitives. Hence, the main innovation in our work (see Equation (5)) is the ability of inferring a synchronous alignment only through attention weights using positions p_t.

Motion encoder To avoid overfitting in the experimented relatively small datasets, we use a low paramter count MLP (Multi-Layer Perceptron) as the frame level feature extractor. Given a motion $x \in \mathbb{R}^{T_x \times 63}$, the MLP encodes the sequence motion on a per-frame basis and produce the encoder outputs $s_x \in \mathbb{R}^{T_x \times d_{enc}}$ where d_{enc} is the dimension of the encoded vector. Here, d_{enc} is the same as the decoder hidden size ($d_{enc} = n'$).

The MLP network consists of k successive non-linear layers as defined by Equation (9), where $f_i(x(t)) = \tanh(W_i.x(t) + b_i)$. The learnable parameters are $W_i \in \mathbb{R}^{63 \times d_i}$ and $b_i \in \mathbb{R}^{d_i}$, $\tanh$ is the hyperbolic tangent, and where L is the number of layers.

$$s_x(t) = (f_0 \circ f_1 \circ \cdots \circ f_{L-1})(x(t)) \qquad \forall t \in [\![0, T_x - 1]\!] \tag{9}$$

4 Training strategies

In previous works by Plappert et al. Plappert et al. [2017] and Inamura et al. Goutsu and Inamura [2021] the Cartesian coordinates are not used, but we hypothesize that their use is more suitable to generate descriptions pertaining to the global motion and its automatic segmentation.

Loss definition We define loss as the usual cross entropy by the Equation (10), where B is the set of training examples within in a given batch, for training we use Adam optimizer with an initial learning rate 10^{-3}. To train, we use Teacher forcing strategy using a ratio of 0.5 that was empirically selected.

$$Loss = -\frac{1}{\mid B \mid} \sum_{(x,y) \in B} \frac{1}{\mid y \mid^{\beta}} \sum_{k=0}^{|y|-1} log(P(y_k \mid y_{<k}, x)) \tag{10}$$

Word Embedding We choose to learn the embedding of words in the training, instead of using a pre-trained model. Usually, pre-trained models produce high dimensional vectors, increasing the number of network parameters and leading to over-fitting for small datasets, which isn't adapted in our context.

Hyperparameter tuning In all experiments, we use grid-search to optimize hyperparameters using ray tune Liaw et al. [2018]. We found that in most cases the best choice for better generalization was lower embedding size (64) and hidden size (64) with number of layers (1), we use also the teacher force strategy with a 0.5 ratio. These parameters are consistent with the small size of the KIT-ML dataset. We run experiments for $L \in \{2, 4\}$ and $d_i \in \{64, 128, 256\}$ except for the final layer $(i = L - 1)$ where the output dimension d_i is fixed as $d_i = d_{enc}$.

5 Experimental Evaluation and Results

The preliminary experiments and observations give us some insights into the effect of various architectural and formulation choices, but a formal and systematic validation of the proposed systems is indispensable. In this section, we first discuss the typical metrics for text generation systems and propose a simple *semantic similarity score* to complement the standard BLEU4 metric. Subsequently, in Experiment 1, we study the effect of the classical soft and local attention formulations compared to our proposed recurrent formulation. Retaining the better suited recurrent formulation of attention, in Experiment 2, we evaluate the improved encoder architecture along with a more advanced decoding method (beam-search).

5.1 Evaluation metrics

Bias in model evaluation using BLEU score While BLEU is a standard metric, it only captures surface correspondence between the references and generated text. In the dataset, a sequence of motion is mapped to multiple references. If we take the case of two samples which have the same sequence of motion approximately and different sets of references. Considering as example the prediction *a person wipes a table.*, and it references *a human wipes on a table.*, *a person stirs with their left hand.*. The prediction is counted correct only if the words are the same. A description with the same meaning and a different formulation will have low BLEU, which doesn't correlate with human judgment. Consequently, the BLEU score doesn't compare semantic correspondence, to overcome this limitation we propose a semantic similarity metric, invariant under paraphrase, to evaluate the semantic adequacy of generated text compared to references.

Sentence semantic similarity score We can devise a simple transformer-based metric, similar to a triplet-loss approach: by exploiting a sentence transformer "Reimers and Gurevych [2019] model trained for paraphrase detection using triplet loss, we can compute a normalized cosine similarity score between each reference and the generated sentence. Since movement descriptions are composed of very common general-domain vocabulary, a generalist sentence

embedding model trained to assign high scores to vector embeddings of paraphrased sentences, can be expected to perform very well at evaluating the semantic equivalence of the generated sentence with regard to the references.

$$score(RC, P) = \frac{1}{\mid RC \mid} \sum_{i \in |RC|} sim(RC_i, P_i) \tag{11}$$

$$sim(RC_i, P_i) = \frac{1}{\mid P_i \mid} \sum_{p \in P_i} \max_{r \in RC_i} sim_{cos}(emb(p), emb(r)) \tag{12}$$

Where RC is the reference corpus and where each $r_i \in RC$ is a set containing multiple references, and where P is the predictions and where $p_i \in P$ is a set containing one or more predictions. emb represents the sentence-embedding model and returns an embedding vector for a sentence or phrase and sim_{cos} is the normalized scalar product between the vectors. Since references can be quite different from one another, although both are correct, we only keep the highest score for a prediction with relation to all corresponding references. Here, the sim_{cos} is used because the sentence transformer paraphrase model used is trained using the cosine similarity as a regression loss. Other sentence-transformers models may require the use of different metrics.

5.2 Experiment 1: Attentions effect using (Bi)GRU-as encoder

Soft attention

We evaluate and compare the angle-based text generation model with the Cartesian coordinate generation models, both bidirectional and unidirectional. Fig.**??** shows the results for both inputs type. The angle joints in this split have allowed to attain a BLEU-4 of 26.3%, while the use of cartesian coordinates has allowed to increase this performance up to 28.6% on the test set ($+2.5\%$). Regarding the semantic evaluation presented in Table 2, the difference is small, between predictions based on joint angles 78.90% versus cartesian coordinates 78.84%. We can confirm this in our error analyses. We note also that in most cases the predictions have a good semantic score ($> 78\%$) while the BLEU score is under 32%, which doesn't represent a fair and realistic evaluation of generation quality or performance (see section 5.1).

Local and recurrent attention

Table 3 presents the results of applying local attention and the extended recurrent p_t computation. The highest BLEU score of 31.1% is obtained for BiGRU + masked local recurrent attention. However, as demonstrated in section 3.1.2, this model does not perform any proper semantic segmentation contrarily to the case of a unidirectional GRU, obtaining a lower BLEU but a higher similarity score 80.54% and showing a medium qualitative synchronization between motion and language. For local attention, the GRU seem to be the better model in terms of BLEU score 30%.

5.3 Experiment 2: Efficacy of MLP-GRU

In this section we discuss the results resulting from using the simple MLP as the encoder network (cf.figure 1 design **3**) and applying local recurrent attention. Table 4 summarizes the BLEU and similarity scores for all $MLP - GRU$ experiments. In all reported scores from the experiments, the hidden size is fixed at $d_i = 64$. The results for a higher hidden size $d_i \in \{128, 256\}$ were slightly lower and sometimes give approximately the same performances as $d_i = 64$, but start to overfit rapidly. There is more marked qualitative difference through visual inspection of a sample, which was globally lower in the case of high dimension. Maximal BLEU and faster convergence was achieved with no mask and a high hidden size. With a mask and a lower hidden size, the training takes more time but the semantic semantic segmentation is better.

Table 2: GRU-based system : Results of soft attention

System	**Soft.att**	
	Similarity	**BLEU score**
GRU-Angles	79.50	28.5
BiGRU-Angles	78.90	26.3
GRU-Cartesian	**79.56**	**31.8**
BiGRU-Cartesian	78.84	28.6

Table 3: GRU-based system : Results of local attention and recurrent attention

System	Mask (D=5)	Similarity		BLEU score	
		Loc.rec.att	Loc.att	Loc.rec.att	Loc.att
GRU-Cartesian	True	78.84	78.44	30.2	**30.0**
GRU-Cartesian	False	**80.54**	78.34	29.7	29.3
BiGRU-Cartesian	True	78.64	79.53	**31.1**	29.3
BiGRU-Cartesian	False	79.31	**79.58**	26.3	27.6

Table 4: Results of MLP with 2 layers, except for the case $(L = 4)$ we use 4 layers.

Design	Mask	D	BLEU	Similarity
MLP-GRU (L=4)	True	5	**32.1**	78.29
MLP-GRU	True	5	**31.6**	**78.87**
MLP-GRU	False	5	28.8	78.29
MLP-GRU	True	9	29.1	78.52
MLP-GRU	False	9	28.4	76.38

Effect of the Mask window

When the mask is enabled, a truncated Gaussian window is applied in each decoding step t. When the mask is disabled, no truncation is performed, and the alignment is bad in several experiments (see section 6.1), along with a lower BLEU score ($< 29\%$) as reported in the Table 4. The alignment wasn't possible without applying the mask, which acts as regularizer during training and reduces overfitting. The strict limitation of visible encoder outputs along with frame-level feature extraction pushes the network to forcibly learn a precise position only through a semantic motion segmentation based on the text. Another important factor is the D value: for $D = 9$ applying the mask doesn't affect the alignment, and in general when D is high, making has almost no effect.

The results of designs *(1,2)* (figure 1) presented in section 5.2 rely more on global information as a simple way to give a correct prediction instead of learning motion segmentation (especially for short motion action). We explored also a *causal* window given a mask range $[\![p_t - D, p_t[\![$ with $D = 5$, but this led to a deteriorated BLEU score.

Action	References	Predictions (beam size=3)
Kneeling	Human ducks down and takes her hands over her head	a person lies *on all fours* and stands up a person lies *on the ground* and stands up a person lies *on the floor* and stands up
Stomping	Human is lifting his leg	a person is stomping with the right foot. a person is stomping with the left foot. a person is stomping with his right foot.
Squatting	A person stretches their arms out and performs *a single squat* A person does some squad exercises	a person performs a squat a person performs a squat squat a person performs a single squat
Walking	Slow walking motion A person walks forward slowly Human slowly goes forward	a person walks 4 steps forward a person walks *slowly* 4 steps forward a person walks forward
Wiping	A human wipes on a table A person cleans a surface making circular motions with their left hand	a person wipes something in front of it with the left hand a person wipes something in front of it a person wipes something in front of him
Waving	A person raises his right arm in front of his face then starts waving Human performs a waving motion with right hand A person waves with its right hand	a person waves with his left hand a person waves with his right hand a person waves with the left hand

Table 5: Predictions of model MLP-GRU (D=5) with a beam size of 3.

Impact of Beam-search

In this followup experiment, we study the impact of beam search on the BLEU@4 score and average sentence length for the MLP-GRU. In Table 5 we show predictions from different actions: we can see that the model has learned a coarse classification of different meanings *(on all fours, on the ground, on the floor*). For the *Stomping* and *Waving* actions we see that the first and second prediction are focusing on which body part is executing the motion (left or right/ leg or hand) but the first prediction is correct. For the *Walking* action, we see that the information about the speed of

motion is only present in the second prediction. We confirm that applying the beam search for different beam size does not give an amelioration of the BLEU score for the models based on MLP. More results are reported in the Table 6. The best BLEU score is already obtained without sampling, thus no need to perform a beam searching in the prediction process, which is expensive in time computation.

5.4 Global comparison

In summary, across the experiments, we observe that models based on recurrent encoders more directly rely on global information. The training for such architectures is easier and converges more quickly to higher BLEU scores. However, the distribution of attention weights along the frames axis doesn't match the intuitive semantic segmentation produced by human analysis. Introducing recurrent local attention and using a simple MLP as an encoder pushes the network to improve the BLEU score only by learning a correct synchronized semantic segmentation. The convergence for such designs takes longer as the decoder is limited to the visible encoder outputs (finding the correct position p_t takes longer).

5.5 Comparison with SOTA

The comparison was challenging because different authors use different seeds to split the dataset (although proportions are the same), without reporting precisely which random seed is used. In addition to reporting the performance of SOTA systems, we also evaluate our models with three different seeds to verify that the variations of performance due to different random seeds is minimal (less than 0.01 difference), which allows some comparability with SOTA systems. The results are reported in the Table 7. Across all random splits, our BLEU score is higher than reported systems by a margin significantly larger than seed-related variations.

5.6 Experiment 3: Semantic Segmentation Evaluation

For qualitative evaluation of primitive segmentation, we label some representative samples in the dataset for every action (*Kicking*, *Walking*, *Stomping*...). For evaluation, we need an adapted alignment score, this process required to introduce precise definition of a motion segment. As detailed in Lin et al. Lin et al. [2016], the definition is specific to the task. In our work, we want to evaluate both alignment and synchronization between description words and motion primitives. For this purpose we propose three adapted method to calculate a segmentation score *i) based on intersection over union, ii) based on intersection over prediction and iii) based only on alignment position*. The main goal of this study is to quantitatively verify the accuracy of synchronization between correctly generated descriptions and motions: we chose to not evaluate the synchronicity for a completely wrong descriptions, which can never be synchronised. The evaluation of synchronicity is a process of verification that comes after a correct semantic prediction.

5.7 Results on HumanML3D and augmented KIT-ML

To further investigate the generalization capability of our method, we conduct additional experiments on the updated version of KIT-ML, which was augmented by mirroring provided in [Guo et al., 2022a]. Moreover, we adapt the proposed architecture to deal with much larger dataset such as HumanML3D [Guo et al., 2022a].

Training configuration. We use the same split on which we conducted all previous alignment experiments. Later, after our work, an augmented version of the KIT dataset was released by [Guo et al., 2022a]. Consequently, we retrained our models and compared them with the state-of-the-art model released by [Guo et al., 2022b].

KIT-ML [Guo et al., 2022a]. The model hyperparameters remain the same for this augmented version.

Model	**D [Mask]**	**BLEU score (beam size)**				
		1	2	3	4	6
MLP	5 [True]	**31.6**	30.6	30.5	30.7	30.6
	5 [False]	28.8	29.7	**30.7**	30.4	30.7
	9 [True]	29.1	30.4	**30.4**	30.1	30.1
	9 [False]	28.4	**29.2**	28.8	28.9	28.6
Deep-MLP	5 [True]	**32.1**	29.4	29.1	29.1	29.8

Table 6: Impact of beam searching on MLP based models

Random state	Model	BLEU score 1st	2nd	3rd	Avg
0	MLP	0.286	0.272	0.267	0.275
	DeepMLP	0.286	0.298	0.262	**0.282**
35	MLP	0.291	0.274	0.267	0.277
	DeepMLP	0.297	0.268	0.270	**0.278**
42	**MLP**	0.304	0.286	0.265	**0.285**
	DeepMLP	0.256	0.265	0.255	0.259
_	GASSL(LSTM)Goutsu and Inamura [2021]	0.288	0.263	0.257	*0.269*
_	GASSL(GRU)Goutsu and Inamura [2021]	0.262	0.243	0.231	*0.245*
_	BiLSTM Plappert et al. [2017]	0.259	0.237	0.249	*0.248*

Table 7: [Original KIT-ML]Comparison with other models on different level of beam searching

Dataset	**Model**	**BLEU@1**	**BLEU@4**	**CIDEr**	**ROUGE-L**	**BERTScore**	**Segmentation**
KIT-ML	TM2T [Guo et al., 2022b]	46.7	18.4	44.2	79.5	23.0	✗
	Ours MLP+GRU	**56.8**	**25.4**	**125.7**	**58.8**	**42.1**	✓
HumanML3D	TM2T [Guo et al., 2022b]	61.7	22.3	49.2	**72.5**	**37.8**	✗
	Ours|MLP+GRU|	**67.0**	**23.4**	**53.7**	53.8	37.2	✓

HumanML3D [Guo et al., 2022a]. For this larger dataset, the word embedding dimension is set to 128 and the hidden size to $d_i = 256$, while the hidden dimension of the `MLP` with 2 layers have respectively an output dimensions of 512 and 256 (represent d_{enc}) and $D = 10$.

Quantitative comparison vs SOTAs. In the Table **??**, we report BLEU scores for different splits and dataset versions, comparing the Transformer and GRU-based generation. Our model based on the proposed local recurrent attention achieve better performance than the transformer based model `TM2T` and more importantly we preserve the semantic segmentation capability.

5.7.1 Segmentation definitions

First, we will lay a few definitions before delving into the evaluation scores.

Word motion segment The motion segment corresponding to the motion phase associated with the *word* w_i from a generated description will be denoted S_i (as defined by Equation 6).

Primitive motion A basic motion characterized by a unique combination of *action*, *direction*, and *speed*. A global change in the motion direction is considered as a *primitive transition motion*.

Language segment A linguistic description segment is the phrase associated with the word describing one specific action. As actions are almost always described by verbs, the linguistic description segments associated are almost always verbal phrases.

The two process given the motion primitive and the corresponding language segment will be referred respectively as motion and language segmentation. The details of these segmentations will be formulated in the section below.

In all following Equations, we note S^e_{seg} the segmentation score of a sample e, and $\mid E \mid$ the cardinality of the set E. We note by $\mathbb{1}_A$ the indicator function ($\mathbb{1}_A = 1$ if condition A is true and 0 otherwise).

5.7.2 Methods for motion and language segmentation

Language segmentation To perform language segmentation, we first retrieve the words corresponding to the actions from the annotations and calculate their indexes k_m in the prediction. Let w_m be the *action word annotation* of a motion segment, we search for the index k_m of the word w_m in the predicted words then for the next annotation w_{m+1} word and so forth until the final primitive segment is reached, we will write the index of the end token as k_e . As a result of this process, formally, a language segment L_m is given by $\boldsymbol{L_m = \{w_r, r \in [\![k_m, k_{m+1} - 1]\!]\}}$.

Motion segmentation We define P_m as the segment of a primitive motion. The calculation of P_m is given by the concatenation of words motion segments according to index in the set, $[\![k_m, k_{m+1}-1]\!]$ as given by the Equation 13. We recall that the interval of the word motion segment S_i was given in the Equation 6.

$$P_m = \bigcup_{i=k_m}^{k_{m+1}-1} S_i \tag{13}$$

For a qualitative evaluation of motion and language synchronization, we propose to calculate the segmentation score using three different methods.

i) Intersection over Union (IoU) Measure the proportion of intersection between the primitive segment and a reference segment. Based on the IoU we calculate the segmentation score by the Equation 14.

$$S^e_{seg} = \frac{1}{n_s}\sum_{k=0}^{n_s-1} \Vdash_{(\mathrm{IoU_k}\geq\theta)},\ \underset{k\in[0,n_s[}{\mathrm{IoU_k}} = \frac{\mid P_k \bigcap G_k \mid}{\mid P_k \bigcup G_k \mid} \tag{14}$$

we note by n_s the number of motion segment, $m \in [\![0, n_s[\![$ is the motion primitive index, θ is the selected threshold, and k_m computed by the language segmentation. The reference motion segment for an *action word* w_k is the ground truth segment G_k, represent also the kth primitive for a given motion sample. The measure IoU is sensitive to the absolute quality of the ground truth, which isn't always desirable due to variability, which led us to also propose the following measures.

ii) Intersection over Prediction (IoP) We introduce the IoP as a measure for the inclusion proportion of P_k in G_k, so that we have $\boldsymbol{IoP_k = 1 \Leftrightarrow P_k \subset G_k}$, meaning that the segmentation is counted as completely correct when the prediction segment interval is in the range of the ground truth interval. This measure correlates well with the perceived visual quality of transcription generation and reduce the impact of annotation subjectivity (start/end primitive annotation uncertainty). In this case, S^e_{seg} is calculated as in Equation (15).

$$S^e_{seg} = \frac{1}{n_s}\sum_{k=0}^{n_s-1} \Vdash_{(\mathrm{IoP_k}\geq\theta)},\ \underset{k\in[0,n_s[}{\mathrm{IoP_k}} = \frac{\mid P_k \bigcap G_k \mid}{\mid P_k \mid} \tag{15}$$

iii) Element of Evaluate if the alignment position p_i is an element of the annotated segment interval. The score S^e_{seg} is directly defined by (16). This is the least strict method counting any amount of overlap as correspondence.

$$S^e_{seg} = \frac{1}{n_s}\sum_{m=0}^{n_s-1}\sum_{i=k_m}^{k_{m+1}-1} \Vdash_{(p_i\in G_i)} \tag{16}$$

Segmentation process The two processes of language and motion segmentation are illustrated in Fig.5. Given a motion x and the predicted description y_x as *"a person walks forward then turn around and walks backs* `<eos>`. Let $\{S_i, i \in [\![0, 10]\!]\}$ be the set of motion segments associated to the eleven words forming the description y_x. Using the definitions above, we have three mappings :
$P_0 \rightarrow\ S_{k_0}, ..., S_{k_{0-1}} \rightarrow$ *walks forwared then* ; $P_1 \rightarrow$ *turns around and* ; $P_2 \rightarrow$ *walks back* `<eos>`.
The indices k_m are found by searching the ground truth word describing the action in the predicted sentence, and then using the corresponding segments for the group of words describing that part of the (figure 5). For this specific example, $(k_0 = 2, k_1 = 5, k_2 = 8)$. Consequently, the motion x is a sequential composition of three primitive motions P_0, P_1 and P_2. Ideally $G_i = P_i$ but the subjectivity of human annotation introduces an inevitable variability in the start/end timestamps. Note that we always include the end token `<eos>` in the final language segment, because it's relate to the end of motion.

5.7.3 Quantitative evaluation

We have manually annotated 68 samples from KIT-ML with *motion segments* and their *action words*. For a compatible comparison between models, we select the samples with a semantically correct prediction in common across the models. This selection results in $N = 35$ common correct predicted representative samples for the calculation of scores.

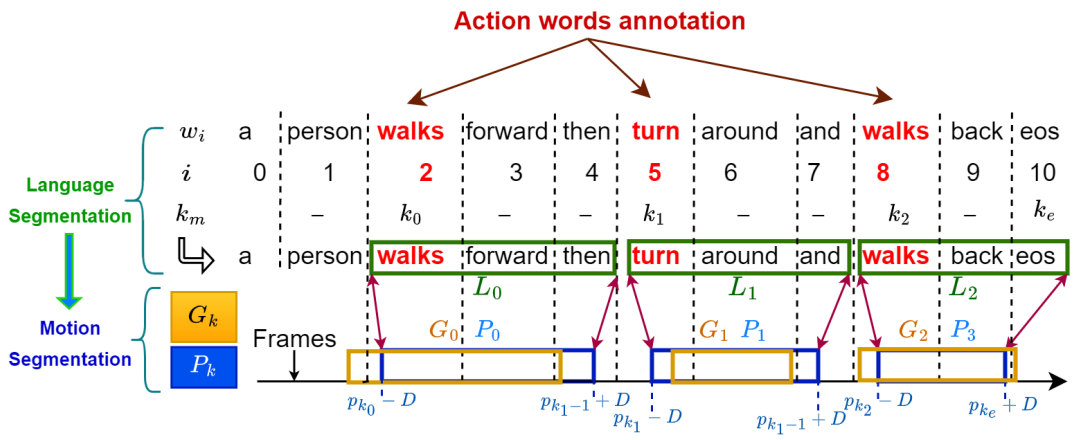


Figure 5: The language and segmentation process applied on an example of predicted words w_i with given action words and frame annotations G_k. The inferred motion segment is P_k and the language segment is L_k. The index k_e always refers to the end token.

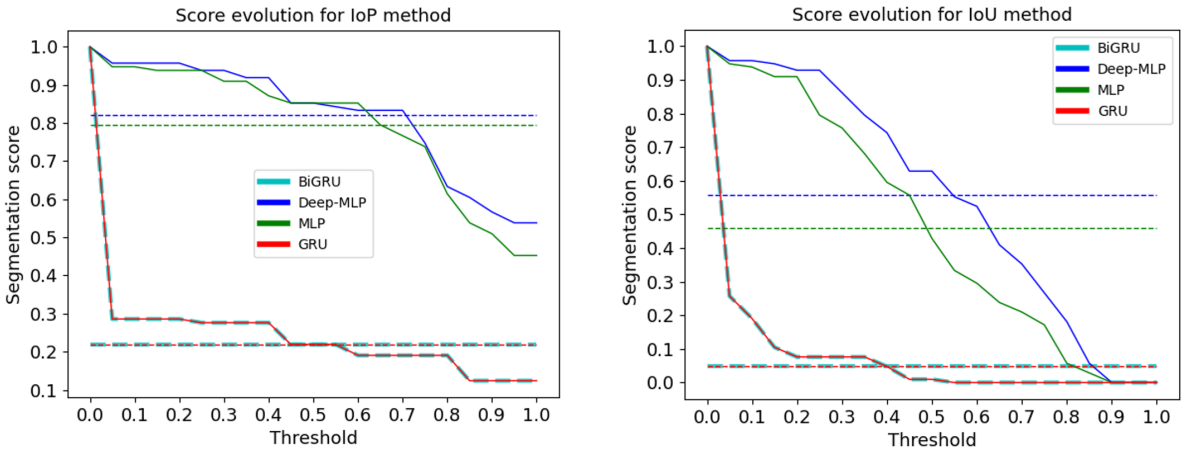


Figure 6: Comparison of IoP and IoU scoring methods. The legend specifies the encoder types. The dashed lines represent the continuous average value without applying the threshold.

Each score is the mean value calculated on all samples $\frac{1}{N}\sum_{e=0}^{N-1} S^e_{seg}$. The scores as defined previously are computed with a threshold, anything below is zero, anything above is one. We can also compute a continuous variant, where we sum the actual scores without thresholding: $S^e_{seg} = \frac{1}{n_s}\sum_{k=0}^{n_s-1} m_k$. Fig.6 illustrates the thresholded (solid lines) score for thresholds from 0 to 1 as well as the continuous scores (dashed lines). *Element of* is not included as it isn't compatible with such a visualisation. As expected the IoU is too strict and the score is almost always zero even when the synchronisation is perceived as correct through visual inspection. We conclude that this measure is not adequate to give a fair assessment of how humans perceive synchronisation. We also observe that the continuous scores remove the parameter that is the threshold and give an informative assessment of synchronisation quality.

In Table 8, we report all continuous scores, including *element of* for a final comparison. We see that the segmentation performance of the GRU was low in comparison with the MLP encoder. The BiGRU has approximately the same performance as GRU. The two have a problem of a late detection of motion. The deeper MLP seems slightly better in segmentation than the shallow MLP regarding the continuous values IoU and IoP. Which is confirmed by the respective curves (figure 6). Noting also that the curves for GRU and BiGRU decrease exponentially w.r.t the threshold, confirming the low ability of recurrent encoders to perform semantic segmentation. Regarding the metric based on *element of*, the MLP encoder is slightly better than Deep-MLP. This means that the MLP encoder localizes the time where the action happens better, while Deep-MLP localizes the start/end of the complete action better.

System	IoU	IoP	Element of
MLP-GRU	45.97	79.47	**89.05**
MLP-GRU (Deep)	**55.92**	**82.06**	88.09
GRU-GRU	5.01	21.90	6.67
BiGRU-GRU	5.01	21.90	6.67

Table 8: Continuous segmentation scores for four encoder type [MLP, Deep-MLP, GRU, BiGRU]. All with Cartesian coordinates and loc.rec.att [D=5,Mask=True]

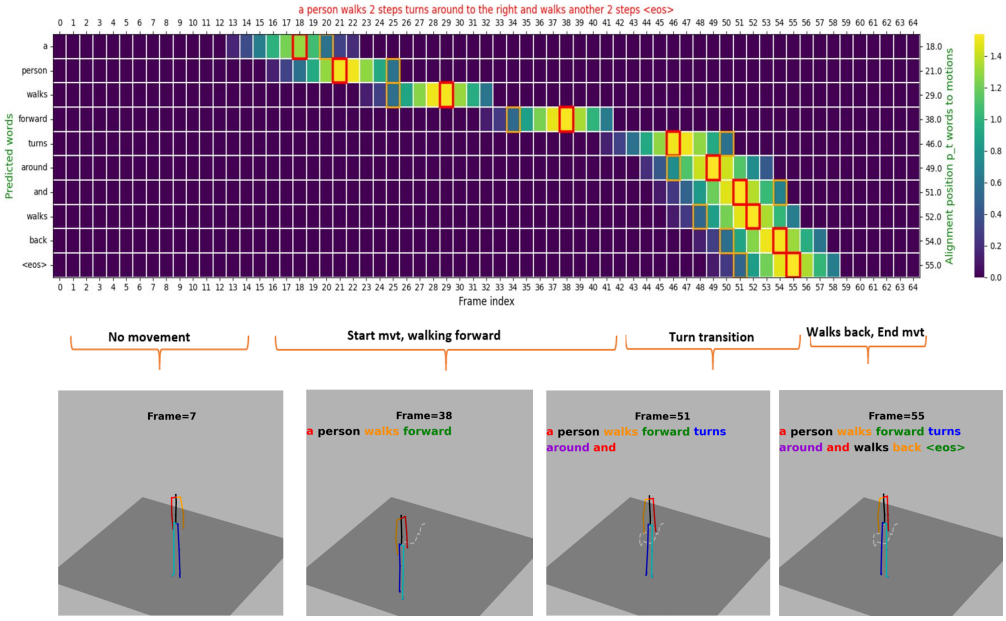


Figure 7: MLP-GRU [Local.rec.att Mask True $D = 5$]: walk forward in the range $[\![15, 38]\!]$, turn at frame 40 , walk backward in $[\![42, 53]\!]$. Another animation illustrating synchronization between text and motion can be found in our code repository.

6 Visual analysis

6.1 Mapping words attention and motion

In this part, we will discuss the perceived synchronization of generated words with motion through a qualitative error analysis. We illustrate this using the evolution of human skeleton motion and attention maps. The Fig.7 shows a correct segmentation in the presence of multiple primitive motions (U-turn). This motion is composed of three primitives "walks forward", "turns around" then "walks back", the primitive "turns around" represents the transition action.

The Fig.8 shows a stomping action, where the word *"stomping"* is associated to the frame interval $[\![22, 30]\!]$, which is the exact moment of execution of the stomping action, followed by an accurate recognition of the body part executing the motion *"left foot"*. Note that the reference and the generated descriptions are perfect paraphrases, they convey the same meaning but use different words. The BLEU score would be incorrectly low in this case. Also note that the attention weight start to be higher when the human skeleton starts moving around the frames in the range $[\![13, 16]\!]$.

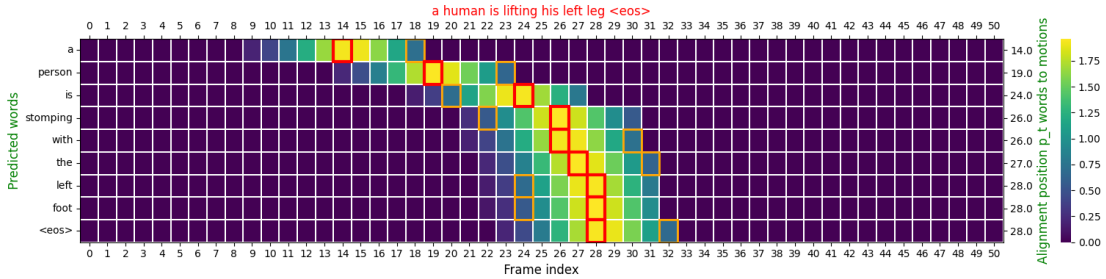


Figure 8: Truncated Gaussian MLP-GRU $D = 5$, Stomping action at $[\![22, 30]\!]$.

6.2 Synchronization between motion and words

We propose a better visualization method for the synchronization by a parameterized skeleton transparency using attention weights. A sequence of motion frame $j \in [\![0, T_x - 1]\!]$ is generated with this transparency values for each predicted word w_i, where the transparency V_{ij} for a frame j given a word w_i is computed from the attention weights using Equation 17. The motion sequence is shown for action words, as illustrated in Fig.9.

$$V_{ij} = \frac{1}{\max_{j<T_x}\{G_{ij}\}} \times G_{ij} \qquad G_{ij} = \frac{\exp\left(F \times \alpha_{ij}\right)}{\sum_{j=0}^{T_x-1} \exp\left(F \times \alpha_{ij}\right)} \tag{17}$$

The factor F is set to 100 for better visualization and controls the number of frames present per image. Note that none of these operations change the position of alignment and maximum of attention, it only helps to have a better visualization of synchronization as a frozen sequence inside the interval $[\![p_t - D, p_t + D[\![$ (mask=True). 3D animation can be found in our repository as a point of comparison to see the relevance of these static visualizations. The visualization of sequence skeleton may be only relevant for words directly describing the motion. We have found that for the repeated words *"a person"* it is usually aligned with the start of movement, providing an initial position to predict the coming action.

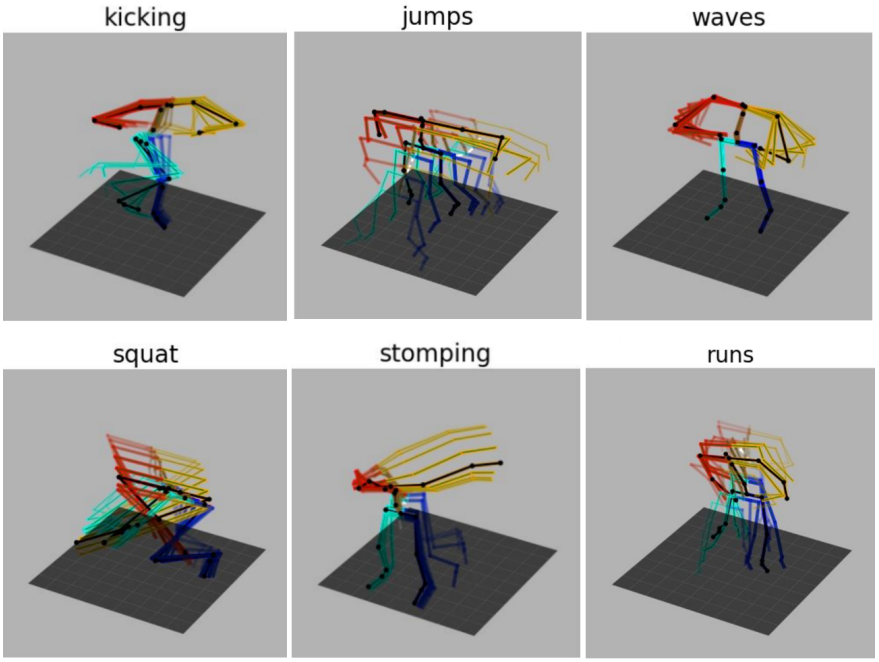
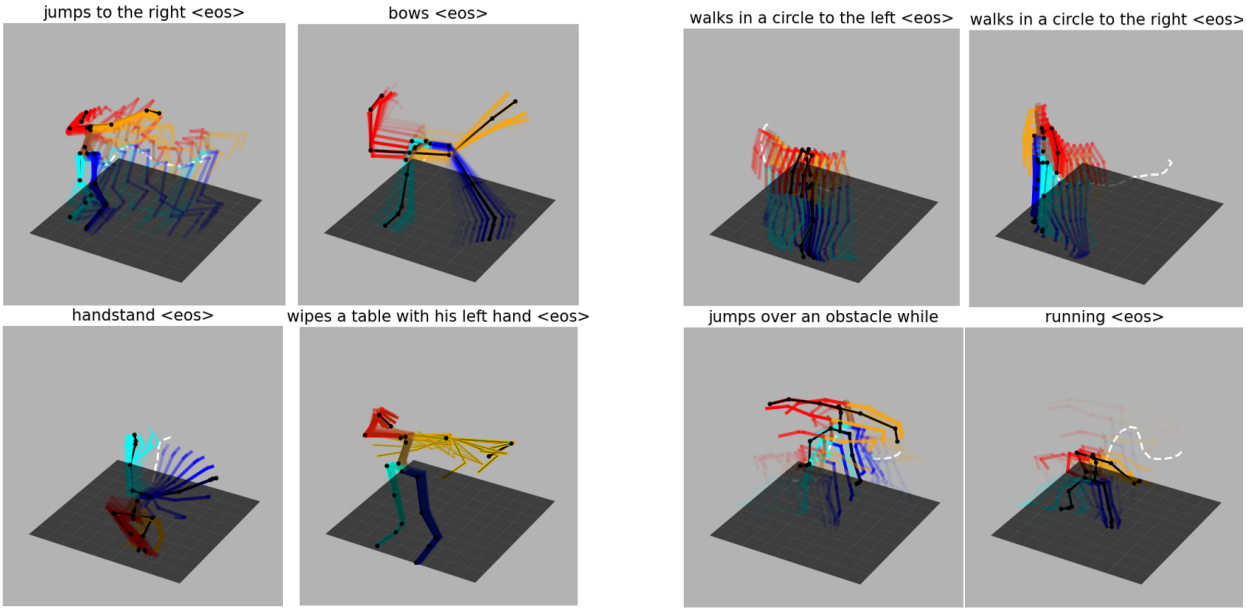


Figure 9: MLP-GRU [Local.rec.att Mask True $D = 5$]: action words and their high attention motion frames. Given a word w_i, the transparency of each frame j is set to value V_{ij} as described in Equation 17 .



(a) Single action.

(b) Multiple actions.

Figure 10: Mapping $P_m \rightarrow L_m$: multiple actions vs. single action.

6.3 Mapping between language and motion segment

As discussed before, it is more pertinent to associate a set of frames not only with a single action word but with the phrase describing the motion with higher granularity, which differs from mere action recognition. The Fig.10 show the mapping of the human pose sequence for the frames in the set P_m described by the language segment L_m. The transparency is calculated as before in Equation 17, but the coefficients α_{ij} are replaced by the coefficients γ_{ij} defined in Equation 18, computed for each language segment L_m ($m \in [\![0, n_s - 1]\!]$).

$$\gamma_{k_m j} = \frac{1}{k_{m+1} - k_m} \sum_{i=k_m}^{k_{m+1}-1} \alpha_{ij} \qquad \forall m \in [\![0, n_s - 1]\!] \tag{18}$$

Specially when $m = n_s - 1$ we should have $k_{m+1} - 1 = k_e$. As described before, we always include the token `<eos>` in the final language segment, so by definition $\boldsymbol{k_{n_s} = k_e + 1}$.

6.4 Limitations of the recurrent attention and dataset

Counting motion The MLD dataset doesn't contain sufficient examples with a variable number of repetitions to learn the invariants necessary for counting repetitions (number of steps, waving hands, . . .). Indeed, in the majority of cases, it includes only a few examples given fixed counts *(walking 4 steps)*. Instead, the architecture needs to see examples with a variable number of repetitions in the training data *(e.g., walks 5,4,2 steps forward)* to allow for a better generalization. However, this is not the only requirement, a strict limitation of visibility of the input in the encoder will not allow the model to learn how to count repetitive motions, as a larger window width would be required. A partial solution can be to apply multiple Gaussian windows that could get counting information.

Dynamic motion measure The information about the speed of the movements was seldom present in the dataset and consequently, the models are rarely able to generate descriptions regarding the speed of movements.

Body part identification We have seen in some samples a wrong detection of the body part executing the motion (e.g. left leg instead of right leg), which is the result of i) few descriptions rich enough to make the distinction in dataset ii) the limitations of the extracted features that do not take skeleton geometry into account. Applying architectures allowing spacial feature extraction can be more pertinent to add this capability.

7 Conclusions and Perspectives

We have proposed a supervised motion-to-language translation system that incorporates non-supervised alignment techniques based on attention weights that allow for synchronous or online text generation. Such a system, beyond the generation of motion descriptions in natural language, can be potentially adapted to other tasks where synchronous generation is important: subtitles of movies, sign language transcription, skeleton-based action segmentation, etc. In general, multiple levers can be leveraged to drive the adaptation to adjacent tasks: adjusting the value of the D parameter, tuning ϵ to control the amount of overlap in the alignments (negative values are permissible). On the tasks of motion-to-language translation, there are additional improvements that may be investigated to further improve alignment performance. We believe that further developments in this area of motion-to-text translation placed in a larger scope, can bring about numerous potential applications, not just in social robotics or human-robot interaction. Motion segmentation approaches on their own have a myriad of potential applications Wang and Rao [2018], but their intersection with description generation permits more focused applications requiring both qualitative description/assessment and time-synchronous delimitation: ranging from sports dialectics and athletic performance optimization to functional rehabilitation medicine.

Data availability statement

The KIT human motion to language dataset used throughout this work is available at `https://motion-annotation.humanoids.kit.edu/dataset/`. The HumanML3D dataset is available at `https://github.com/EricGuo5513/HumanML3D`. Any scripts performing post-processing of said data before it was used, as well as the portion we annotated with synchronization information, and the full implementation of our models, will be made available on our git repository upon the final publication of this paper.

Acknowledgements

The following work is supported by the scholarship granted by the Occitanie Region of France (Grant number ALDOCT-001100 20007383).

References

Christian Mandery, Ömer Terlemez, Martin Do, Nikolaus Vahrenkamp, and Tamim Asfour. Unifying representations and large-scale whole-body motion databases for studying human motion. *IEEE Transactions on Robotics*, 32: 796–809, 8 2016. ISSN 15523098. doi:10.1109/TRO.2016.2572685.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5152–5161, June 2022a.

Matthias Plappert, Christian Mandery, and Tamim Asfour. The KIT motion-language dataset. *Big Data*, 4(4):236–252, dec 2016. doi:10.1089/big.2016.0028. URL `http://dx.doi.org/10.1089/big.2016.0028`.

Matthias Plappert, Christian Mandery, and Tamim Asfour. Learning a bidirectional mapping between human whole-body motion and natural language using deep recurrent neural networks. *Robotics and Autonomous Systems*, 109: 13–26, 5 2017. doi:10.1016/j.robot.2018.07.006.

Angela S Lin, Lemeng Wu, Rodolfo Corona, Kevin Tai, Qixing Huang, and Raymond J Mooney. Generating animated videos of human activities from natural language descriptions, 2018.

Anindita Ghosh, Noshaba Cheema, Cennet Oguz, Christian Theobalt, and Philipp Slusallek. Synthesis of compositional animations from textual descriptions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1396–1406, October 2021.

Mathis Petrovich, Michael J. Black, and Gül Varol. Temos: Generating diverse human motions from textual descriptions. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 480–497, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20047-2.

Yusuke Goutsu and Tetsunari Inamura. Linguistic descriptions of human motion with generative adversarial seq2seq learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4281–4287, 2021. doi:10.1109/ICRA48506.2021.9561519.

Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *ECCV*, 2022b.

Jonathan Feng Shun Lin and Dana Kulic. Online segmentation of human motion for automated rehabilitation exercise analysis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22:168–180, 2014. ISSN 15344320. doi:10.1109/TNSRE.2013.2259640.

Dana Kulić, Wataru Takano, and Yoshihiko Nakamura. Online segmentation and clustering from continuous observation of whole body motions. *IEEE Transactions on Robotics*, 25:1158–1166, 2009. ISSN 15523098. doi:10.1109/TRO.2009.2026508.

Feng Mei, Qian Hu, Changxuan Yang, and Lingfeng Liu. Arma-based segmentation of human limb motion sequences. *Sensors*, 21(16), 2021. ISSN 1424-8220. doi:10.3390/s21165577. URL `https://www.mdpi.com/1424-8220/21/16/5577`.

Rui Li, Zhenyu Liu, and Jianrong Tan. Human motion segmentation using collaborative representations of 3d skeletal sequences. *IET Computer Vision*, 12:434–442, 2018. ISSN 17519640. doi:10.1049/IET-CVI.2016.0385.

Feng Zhou, Fernando De la Torre, and Jessica K. Hodgins. Aligned cluster analysis for temporal segmentation of human motion. In *2008 8th IEEE International Conference on Automatic Face and Gesture Recognition*, pages 1–7, 2008. doi:10.1109/AFGR.2008.4813468.

Feng Zhou, Fernando De la Torre, and Jessica K. Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):582–596, 2013. doi:10.1109/TPAMI.2012.137.

Hao Ma, Zaiyue Yang, and Haoyang Liu. Fine-grained unsupervised temporal action segmentation and distributed representation for skeleton-based human motion analysis. *IEEE Transactions on Cybernetics*, 2021. ISSN 21682275. doi:10.1109/TCYB.2021.3132016.

Benjamin Filtjens, Bart Vanrumste, and Peter Slaets. Skeleton-based action segmentation with multi-stage spatial-temporal graph convolutional neural networks. *IEEE Transactions on Emerging Topics in Computing*, pages 1–11, 2022. doi:10.1109/tetc.2022.3230912. URL `https://doi.org/10.1109%2Ftetc.2022.3230912`.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, pages 1–15, 2015.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, 2015. Association for Computational Linguistics. doi:10.18653/v1/D15-1166. URL `https://aclanthology.org/D15-1166`.

Hannah Bull, Michèle Gouiffès, and Annelies Braffort. Automatic segmentation of sign language into subtitle-units. In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision – ECCV 2020 Workshops*, pages 186–198, Cham, 2020. Springer International Publishing. ISBN 978-3-030-66096-3.

Hannah Bull, Triantafyllos Afouras, Gül Varol, Samuel Albanie, Liliane Momeni, and Andrew Zisserman. Aligning subtitles in sign language videos. *ICCV*, pages 11552–11561, 2021. URL `https://www.robots.ox.ac.uk/`.

Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E. Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training. *CoRR*, abs/1807.05118, 2018.

Nils "Reimers and Iryna" Gurevych. "sentence-BERT: Sentence embeddings using Siamese BERT-networks". In *"Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)"*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi:10.18653/v1/D19-1410. URL `https://aclanthology.org/D19-1410`.

Jonathan Feng-Shun Lin, Michelle Karg, and Dana Kulić. Movement primitive segmentation for human motion modeling: A framework for analysis. *IEEE Transactions on Human-Machine Systems*, 46(3):325–339, 2016. doi:10.1109/THMS.2015.2493536.

Qifei Wang and Yunbo Rao. Visual analysis of human motion: A survey on recent advances and applications. In *2018 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4, 2018. doi:10.1109/VCIP.2018.8698618.

Appendix A

In all conducted experiments the common observation is that most motion descriptions start with the words “a person”, for such words that depend on language model, the network learns to align this word with the start of the motion.

Fig.11 shows the case of a motion composed of two motion ’move to the left’ then ’move to the right’ where localization of attention weights was correctly distributed allowing a correct segmentation of motion.

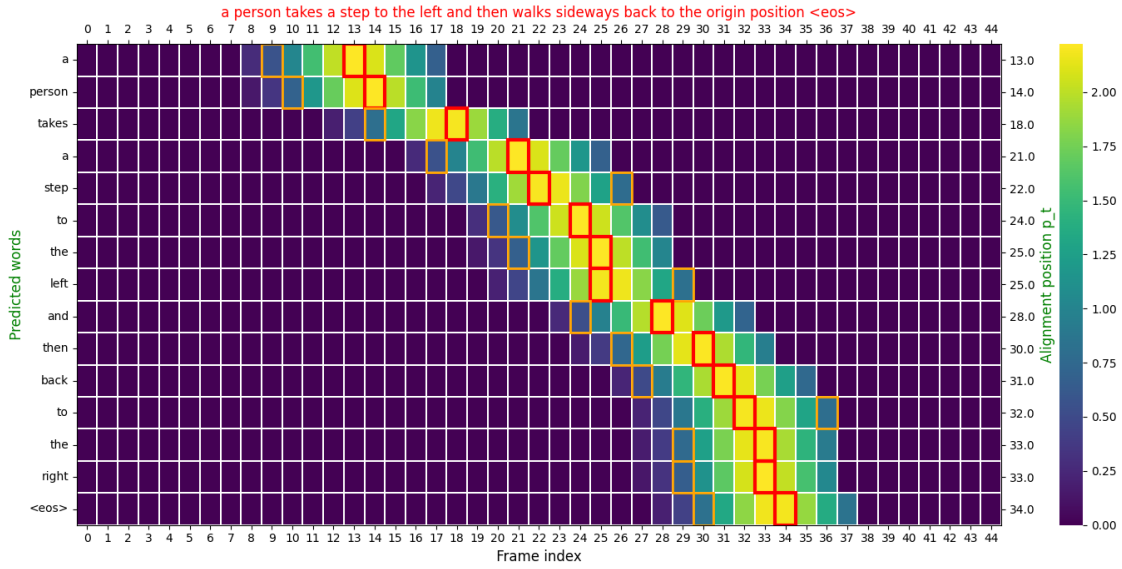


Figure 11: Truncated gaussian DeepMLP-GRU $D = 5$ [move left, move right].(move to the left in the range $[\![8, 23]\!]$, move to the right in the range $[\![24, 35]\!]$

Fig.12 shows the case of push backward action, the attention positions were correctly distributed with relation to the range of the action.

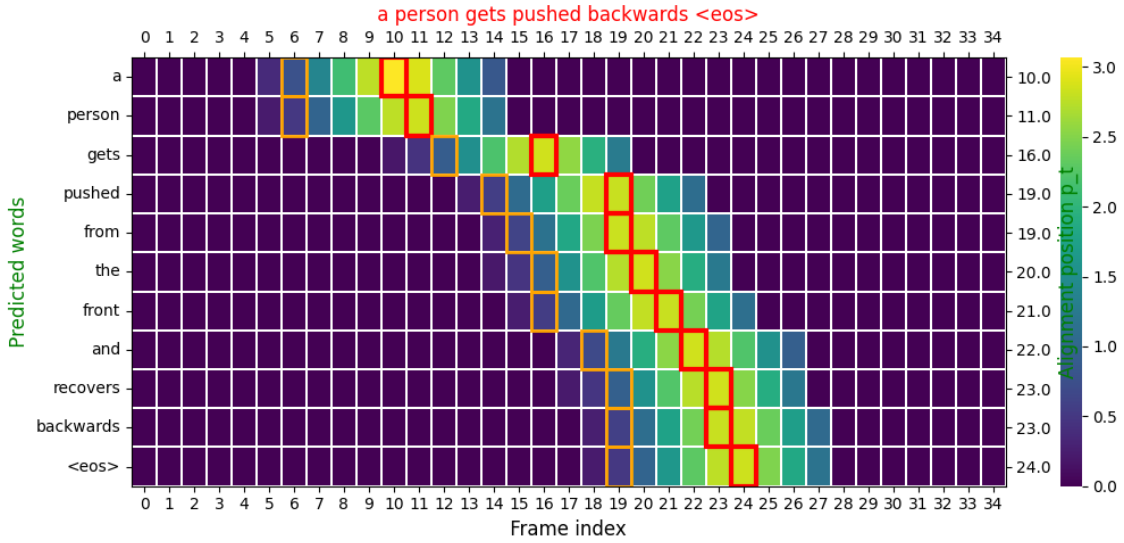


Figure 12: Truncated gaussian MLP-GRU $D = 5$ [Push backward].(pushing action in the range $[\![15, 26]\!]$.

Fig.13 represents the attention map for "walk down", when observing the motion, the action is performed in the range $[\![11, 30]\!]$.

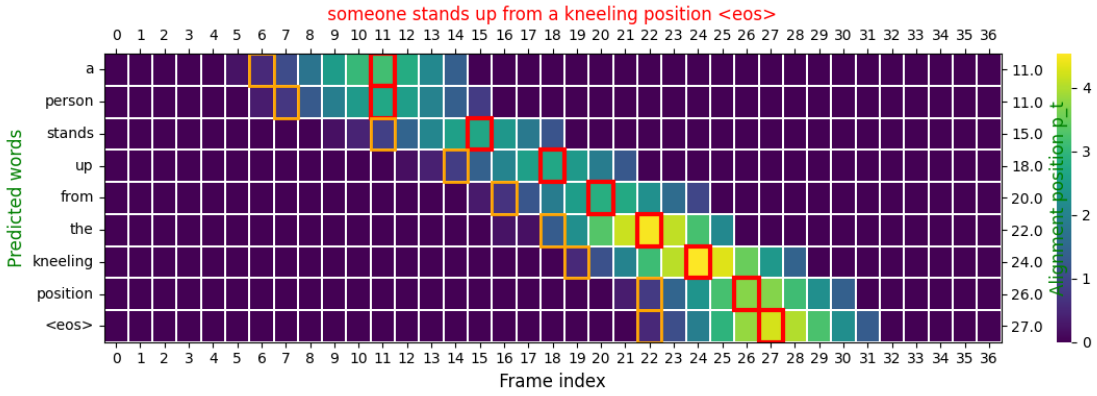


Figure 13: Truncated gaussian DeepMLP-GRU $D = 5$ [stand up].(range $[\![11, 30]\!]$)

Fig.14, presents the association P_m to L_m for model using the deep-MLP encoder.

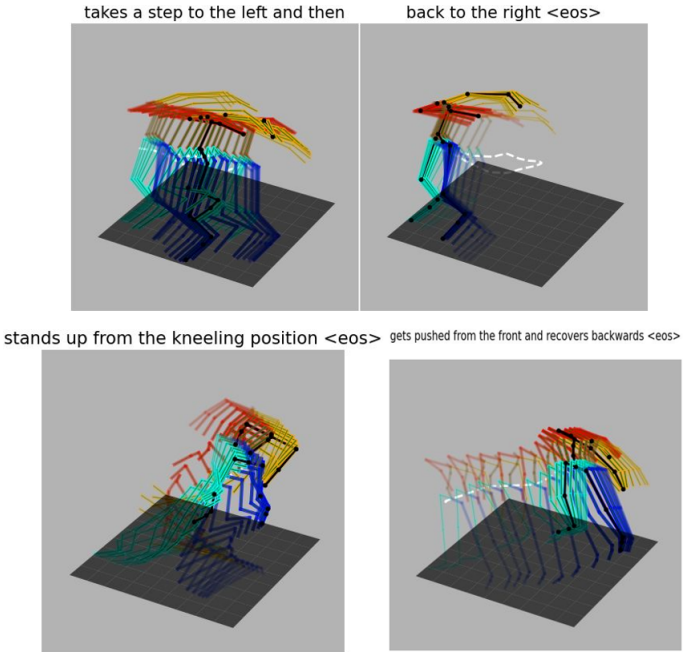


Figure 14: Truncated gaussian DeepMLP-GRU $D = 5$ mapping $P_m \rightarrow L_m$