MDM Motion Generation
Advancing the Perception of Time in Motion

Roy Diamant, Ben Barak, Bar Shaked

1. Problem and Theoretical Background

One of the primary challenges in generating human motion sequences using natural language text is accurately perceiving and representing temporal relationships within the input text. The MDM model, like many other natural language processing models, struggles with this task due to the inherent complexity and variability of human language. There are several reasons why this issue may be particularly pronounced in the MDM model. One possible reason is that the transformer architecture used in the model is primarily designed for sequence-to-sequence tasks, such as language translation or summarization, rather than capturing temporal relationships within a single sequence. Additionally, the diffusion model used in the MDM model may prioritize smoothness and coherence over explicit temporal relationships, which could further contribute to the difficulty of accurately representing temporal relationships in the generated motion sequences. Finally, the use of CLIP for text encoding may also contribute to the difficulty of perceiving temporal relationships, as the model was primarily trained on images rather than text, and may not be optimized for capturing the subtle temporal relationships within natural language text.

As an example of the challenges of perceiving temporal relationships in natural language, consider the sentence "The person jumps and then raises his hands." While this sentence may appear straightforward, it presents several challenges for the MDM model. Without a clear temporal marker such as "before" or "after," the model must rely on subtle contextual cues to determine the correct temporal relationship between the two actions. However, if these cues are ambiguous or difficult to discern, the model may generate a motion sequence where the person raises his hands before jumping, or where he jumps and raises his hands simultaneously. In either case, the generated sequence would not accurately reflect the intended temporal relationship within the input text. This example highlights the potential impact of the problem of perceiving temporal relationships on the generated motion sequences and underscores the need for new approaches to addressing this challenge.

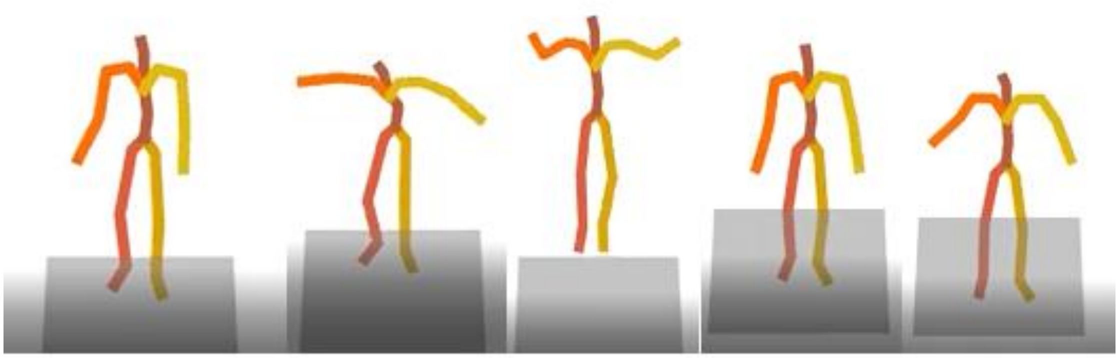

Figure 1. Example of a motion generated by the MDM model based on the input text 'The person jumps and then raises his hands.' In this sequence, the model generates a motion of the person jumping and raising his hands simultaneously instead of jumping and then raising his hands. This highlights the challenge of perceiving temporal relationships in natural language and the need for improved models to address this issue.

2. Suggested Solution

Firstly, in order to improve the accuracy of the Motion Diffusion Model (MDM), we will break down the input sentence into smaller segments based on specific words that indicate the occurrence of events over time. This process will be carried out using the *"temporal text analyzer"*, which will provide us with a more precise understanding of the temporal relationships within the sentence. By breaking down the sentence in this manner, we can ensure that the generated motions accurately reflect the meaning of each segment.

Secondly, we will utilize the "text to motion" feature of the MDM to generate a unique motion for each segment of the sentence. By using this feature, we can generate dynamic and realistic motions that accurately reflect the meaning of each segment.

Finally, we will use the "in-betweening" feature of the MDM to combine all the generated motions in the correct order of the time events. This feature enables us to create a smooth and coherent motion sequence that accurately reflects the meaning and order of the events in the text.

To achieve natural and coherent motion using in-betweening, determining the optimal number of frames to generate between each segment is crucial. Our solution includes a sophisticated function (*'Motion Gap Function'*) that predicts these numbers based on the length and characteristics of each segment, ensuring seamless and consistent transitions between each segment. In the subsequent part of our solution implementation description, we will provide a detailed explanation of this function and its contribution to generating high-quality human motions.

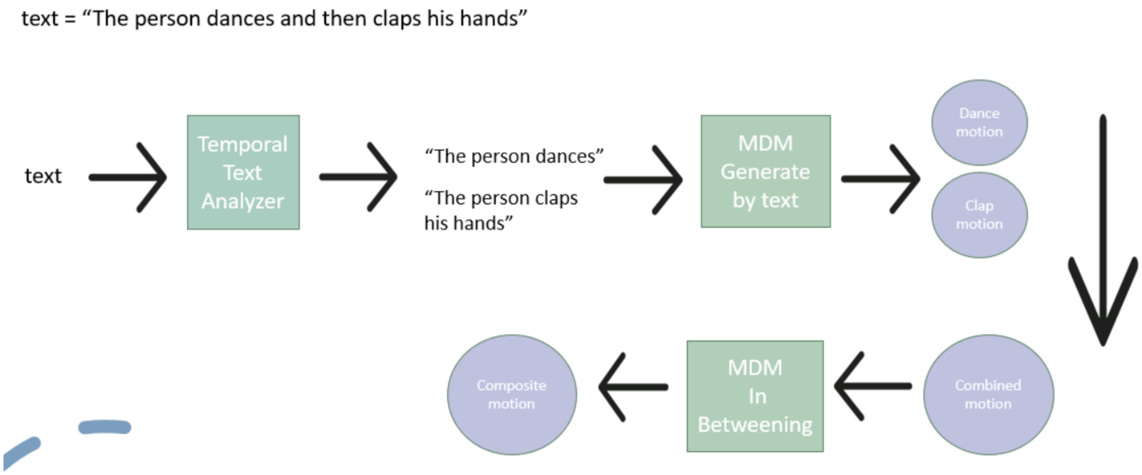

Figure 2. Flowchart illustrating motion generation for the sentence 'The person dances and then claps his hands' using the proposed solution. The sentence is broken down into smaller sentences based on temporal relationships: 'The person dances' and 'The person claps his hands'. The Motion Diffusion Model (MDM) generates unique motions for each segment, i.e., dancing and clapping. The motions are then combined using in-betweening to create coherent motion, while ensuring they are arranged in the correct order of temporal events.

3. Implementation of the solution

3.1 Temporal Text Analyzer

To implement the temporal text analyzer, we wrote a Python script that uses regular expressions to split a sentence into parts based on a set of conjunctions that imply temporal relationships between those parts. The script defines two sets of conjunctions: *'After Conjunctions'* and *'Before Conjunctions'*. The former set includes conjunctions that imply the prefix action is done after the suffix action, while the latter set includes conjunctions that imply the prefix action is done before the suffix action.

The *'Temporal Text Analyzer'* takes a sentence as input and splits it into parts based on the conjunctions. It then reorders the parts based on the conjunctions and removes the conjunctions to return the reordered parts.

3.2 In-Betweening

The original in-betweening feature in MDM is designed to work only with random samples from the dataset, thus we had to extend this functionality to support generating in-between frames for custom motions that are not included in the dataset.

Furthermore, we improved the in-betweening feature by allowing it to work with multiple motions. This means that instead of generating poses only between the start and end of a single motion, we can now seamlessly combine the end of one motion with the start of another.

3.3 Motion Gap Function

We have explored various approaches to determine the MGF, until we arrived at a suitable solution.

3.3.1 The Naive approach

The naive approach to implementing the "Motion Gap Function" involves giving the "in-betweening" feature of the MDM an equal number of frames from the end of the first motion and the start of the second motion, and a fixed number of frames between them to generate. We initially chose to use a large number of frames (60), with the expectation that the "in-betweening" feature would be able to generate a smooth and high-quality motion sequence. The idea is that it will have enough frames for a reasonable transition between any kind of two different motions.

3.3.2 MGF Using Diverse 3D Human Motion Generation Model

For this approach, we used a model described in a paper called “Generating Diverse and Natural 3D Human Motion from Text” (By the University of Alberta). This model gets a text input and outputs the number of frames required to generate the corresponding motion.

To use this approach for our motion gap function, we first break down the input text into its constituent parts (in the case of the example “The person danced and then jumped”, this would be “The person danced” and “jumped”). We then input each of these parts separately into the model and receive the estimated length of the generated motion for each part.

Finally, we use the estimated length of the entire sentence, as well as the length of each individual part, to calculate the length of the motion that needs to be generated for the gap between the two parts.

$$\#frames = \min\left(15, f(entire\ sentence) - 0.25\left(\sum f(curr\ part)\right)\right)$$

For example, if the length of “The person danced” is 80 frames and the length of “jumped” is 60 frames, and the length of the entire sentence “The person danced and then jumped” is 100 frames, then the length of the gap motion would be calculated as follows:

$$\#frames = \left(100 - (0.25 \cdot 80) - (0.25 \cdot 60)\right) = 65$$

Once we have determined the value of this parameter, we must then decide how many frames to extract from each motion. To accomplish this, we use the estimated duration of each motion segment and extract a proportionate number of frames from it. By doing so, we provide the in-betweening feature with a better understanding of the input motions, allowing it to generate a smooth, continuous transition between them in a more natural manner.

4. Results

Our solution effectively addressed the challenge of accurately perceiving and representing temporal relationships within natural language text when generating human motion sequences, demonstrating significant improvements over the original use of the MDM. We found that the proposed solution did an excellent job in most of the cases where the original MDM failed to accurately capture the intended temporal relationships within the input text.

Specifically, the problem in the original MDM model where sentences like "the man jumps and then raises his hands" sometimes generated motion sequences where the person does not raise his hands or does not jump, no longer occurred in the proposed solution.

We tested the temporal text analyzer script with a variety of input sentences to ensure that it correctly identified the temporal relationships between the parts and returned them in the correct order. Overall, we found that the temporal text analyzer was an effective tool for analyzing the temporal relationships between actions in a sentence.

However, we did encounter some cases where the temporal text analyzer was unable to correctly identify the relationships between parts, particularly in cases where the sentence structure was more complex of the conjunctions were less common, Nonetheless, we believe that for the majority of cases where the input sentences are not too complicated, the temporal text analyzer works effectively.

Combined with the function used to determine the optimal number of frames between each segment most of the time resulted in a coherent and natural-looking motion sequence. The combination of these features resulted in motion sequences that were able to capture the temporal relationships within the input text with a high degree of accuracy.

After examining the initial approach (*The naïve approach),* we discovered that in most cases, the transition generated between the two motions was smooth and natural (see the folder named *'example2')*. The large number of frames we allowed the MDM to generate in between seemed to be sufficient for the task. However, there were instances where the large number of frames resulted in a static and uninteresting transition or one that was unrelated to the input text (refer to the folder name by *'example1')*.

Upon examining the second approach, we found that it improved the result in cases where the naïve approach had failed (see the folder named *'example6')*. In those cases, a smaller number of frames generated by the MGF was enough to achieve a smooth transition that was natural with respect to the nature of the two motions. The smaller number of frames also prevented the transition from being too static, boring, or unrelated to the input text. An interesting thing to notice is that in cases where the two motions are semantically similar both approaches did well. However, in cases where the two movements are essentially semantically different, the task is much more complex, and the second approach gives better results.

We also observed that taking proportional parts from each motion improved the result by avoiding the unnecessary generation of frames for motions that can be described using a small number of frames. Moreover, the MGF performed well in the cases where the naïve approach was successful, indicating that it does not harm the results (see the folder named '*example5'*). The numbers generated by the MGF were also close to the number we arbitrarily chose in the naïve approach.

Overall, our results demonstrate that the proposed solution effectively addresses the challenge of generating human motion sequences when given an input text which describes several actions in a temporal relationship.

5. Future Research Directions for Advancing Temporal Perception

5.1 Temporal Text Analyzer

Based on the limitations of the current Temporal Text Analyzer, future research directions to improve the system could focus on developing a more advanced machine learning model based on recurrent neural networks (RNNs). This could involve incorporating attention mechanisms into the model to selectively attend to different parts of the input text when making predictions about temporal relationships. This could help the model to be more effective in capturing complex temporal relationships that may involve multiple events or temporal markers.

Improving the quality and diversity of annotated training data could also be a fruitful area of research, by using more diverse text sources and incorporating more nuanced annotations, such as information about the duration or frequency of events. By developing a more advanced machine learning model and improving the quality of training data, the Temporal Text Analyzer could be enhanced to better handle complex temporal relationships and provide more accurate and informative temporal analyses of text data.

5.2 Motion Gap Function

As demonstrated in our proposed solution, determining the optimal number of frames between segments is critical for generating natural and coherent motion sequences. However, our current function for predicting the number of frames may not capture all relevant factors that influence the smoothness and realism of the generated motions. Therefore, future research can explore several directions to improve this function.

One promising direction is to integrate machine learning techniques to learn the optimal number of frames based on a training dataset of motion sequences and their corresponding text inputs. This approach can enable the function to adapt to different types of input sentences and generate more realistic and varied motions.

We usually want to determine this number of frames only after the motions were generated using a text input, therefore we will have both the text input and the actual frames of the generated motion, to use as parameters for this machine learning-based approach. Several details might need to be considered, the type of motions of each segment and the nature of the transition between them, the starting frames of the second segment and the final frames of the first segment, and the velocity, and length of each of the segments.