# Generating Sign Language Videos using DDPM

## January 26 2023

Nikita Belooussov

Supervisors:
Sergio Escalera
Germán Barquero
Albert Clapés i Sintes

BARCELONA SCHOOL OF INFORMATICS (FIB)
FACULTY OF MATHEMATICS AND INFORMATICS (UB)
SCHOOL OF ENGINEERING (URV)




Barcelona, January 19, 2023

# Abstract

This thesis focuses on neural networks being able to generate a motion array or an array of landmarks that represent a sign language interpreter when given a sentence. This is important due to the possibility of providing more accessibility for people that need to use sign language to communicate. The specific model being used is a denoising diffusion probabilistic model(DDPM), which has recently gained popularity in video, image, motion, and landmark generation. The dataset being used will be the LSFB dataset, which uses French for the input sentences and has videos of people signing in the French Belgian sign language. The thesis will show that it is possible to use these types of models to generate the translations required.



**Key words:** Sign Language, Video Generation, Denoising Diffusion Probabilistic Models

# Acknowledgements

There are several people that I would like to thank for their help with this thesis. First are my academic supervisors Sergio Escalera, Germán Barquero, and Albert Clapés i Sintes, without them I would have never come up with the idea for this thesis or known where to start with it. They have helped immensely with my understanding of the models and have helped me on short time notice. Thank you for your patience with me and all of your help throughout the development of the thesis.

I would also like to thank Jerome Fink for helping me with obtaining and understand the LSFB dataset. Through his help, I was able to obtain a much greater understanding of how the dataset works and how to interpret it.

Lastly, I would like to thank my family. They always provided me with emotional support and provided me with anything that I may have needed to help me finish the thesis. Without them, it would have been much more difficult to complete the thesis.

# Contents

# List of Figures



# List of Tables

# 1 Introduction

As Artificial Intelligence starts to become better, it is important to find ways to not only improve businesses and manufacturing but also help improve people's lives. This thesis aims to show that arrays of motion or landmarks could be generated from simple sentences. These arrays could be later used to produce videos of sign language interpreters. This involves the AI model being able to learn the movements associated with the words and word phrases in sentences. It will also have to understand that the same word may have different interpretations. As a result, it may have various movements associated with it.

There have been many several papers already published surrounding AI and sign language interpretation. These generally break down into two categories. The first is translating from video to sentences by recognizing the signs [3] [27]. The second one, which is the focus of the thesis, is creating videos of a sign language interpreter [37] [26] [18] [36] [32] [31] [30]. This is done generally done with the input being either audio or text. This input is then translated to either a video of an interpreter or to landmarks/motion which can later be used to produce a video of an interpreter. Although, the papers that went from audio to video would generally have the text generated as an intermediate step between the two.

A model that is able to produce a video in sign language video from a given text could be extremely useful. This is mainly due to how many different scenarios it could be applied in and how it will increase accessibility to a portion of the population. This will be a good step for many people that have to use sign language to communicate. The main hurdles that need to be overcome are for the model to be creating the correct motions, and for the people who interact with it to prefer it to other forms of communication that they may have accessible. Only the first part of the problem will be addressed in this thesis, with possible suggestions for the latter only being discussed in the Future Work Section 7. This section will be describing important parts of the thesis.

This thesis will be exploring the use of a denoising diffusion probabilistic model(DDPM) for the described task. These models have recently grown in popularity, especially for video, image, motion, and landmarks generation. The model used in this thesis is one that was created to produce landmarks that correlate to a motion that a text input is describing. Due to how similar the two tasks are, it was decided that this model would be a good starting point for this thesis. There will be more information provided about DDPM models in Section 1.3 and about this specific model in section 4.1.

As mentioned, the goal of this thesis is to create a DDPM model which can be used to create these motion arrays and arrays of landmarks. These arrays later can be used to produce a realistic-looking video of a sign language interpreter. Due to time restraints and restraints on computing power, the problem was significantly simplified. The first was limiting the number of frames to 500. The second was to only produce either the landmarks or motions of certain points of interest on the sign language interpreters. In the

future, a more detailed set of outputs, similar to these ones, can be combined with either other AI models or other programs to help generate realistic videos. The remainder of this section will further cover the basic information required to know for this thesis and give an explanation for certain choices.

## 1.1 Sign Language

The most important part of the problem is sign language. At first, one may assume that sign language is a one-to-one translation, such as going from spoken form to the written form of the same language. This is not the case in sign language. Translating sign language should be approached more as translating between two different languages. This already proves to be difficult, since a word-for-word translation may not always be ideal. The syntax or word order may be incorrect in the translated language. In sign language, the same would apply, especially since it could be possible for separate words or ideas represented in the sentence, to be signed simultaneously.

In fact, due to the differences between sign language and written language, there is often an intermediate step seen in datasets. Since signs often represent ideas and not a specific word, a gloss( a name for the motion being made) is often written. An example can be seen in Figure 1

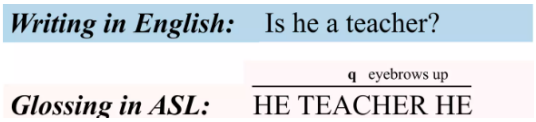


Figure 1: Example of Gloss [1]

As shown in Figure1, the translation is not a one-to-one translation. It is also important to note that in many of the datasets, additional information, such as"eyebrows go up" is missing. The dataset only contains the names of the signs being used. Although for training and testing, this information was not used. Since it is unlikely that this would be the input format in day-to-day use. The input into the model is the normal sentence in French. This shows the increased difficulty of creating the create motions to be able to understand the correct glosses.

Another challenge that arises from sign language is that motion generated from person to person can be different. This can be due to the speed at which the motion is happening, the dimensions of the interpreter, motions appearing to be different due to different camera angles, etc. All of these issues mainly arise in training the model and are things that the model will have to overcome.

## 1.2 Dataset

In this thesis, the dataset that will be used is the LSFB Continuous dataset [8]. This dataset uses the French Belgian sign language. This means that both the written language and the sign language will be in French. This is due to the possible additional errors that could be introduced when translating the sentence inputs in the dataset to another language. The original purpose of the dataset was to train models to translate from a video of sign language to a text output, but the dataset can be used in the opposite direction as well.

There are other datasets that are often used with sign language tasks, such as the RWTH-PHOENIX-Weather [19] or the MS-ASL [38] datasets. Both of these were discarded for various reasons. For example, the MS-ASL dataset only provided videos for single words and not entire sentences. For this thesis, it was decided that the entire sentence should be translated and not only a single word. In order to better show the possibility of a full translator being possible. The RWTH-PHOENIX-Weather dataset does allow for translations of entire sentences. The dataset is based on the weather forecasts done on a news channel. It also provided the landmarks that were required for the thesis. All of these benefits are also provided in the LSFB Continuous dataset as shown in Fig 2.

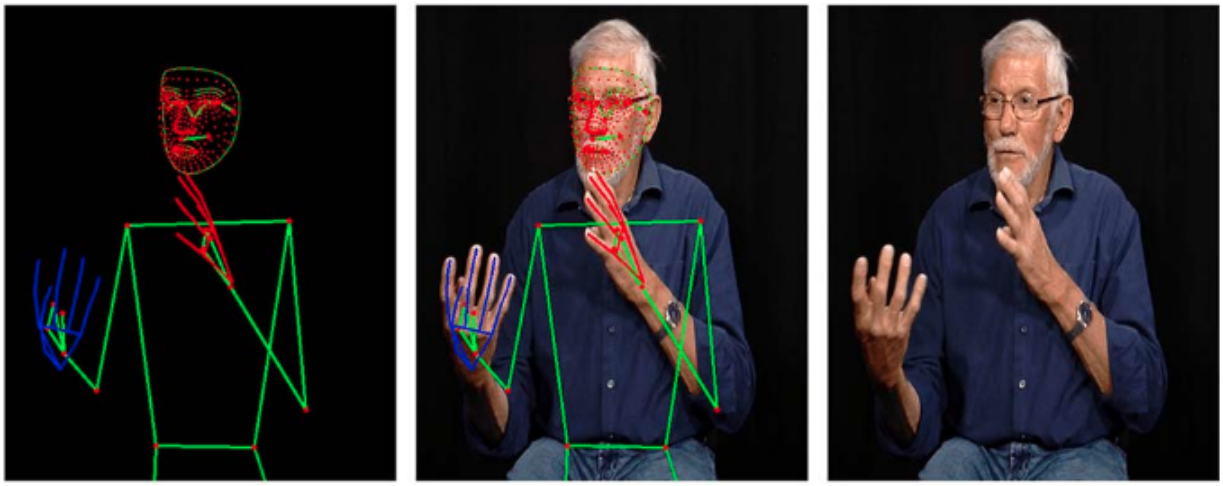

Figure 2: Example of Landmarks in LSFB [8]

As mentioned, one of the reasons that the LSFB Continuous dataset was chosen is due to it containing the landmarks that are tracked throughout the videos. The other reason for using this dataset was its size. The dataset had more than 85,000 sentences to train on. A large dataset was required due to the filtering that was applied to the dataset. This filtering will be described more in-depth in Section 4.2. A more in-depth discussion of the datasets and the choices made about them will be in Section 4.2.

## 1.3 DDPM

The goal of the thesis is to prove that a DDPM model would be able to solve this task. For this reason, it is important to understand how a DDPM model operates. The main paper [14] on this model was written in 2020. Since its publishing, the model has quickly gained popularity and has been used in many different scenarios. This varies from working with simple images [2] [29], to working with videos [15] [45] [13] [40] [16] [47], and working with motion and landmark generation [4] [6]. Due to its popularity and how new the model is, it was decided that this would be a good model to test as a solution to this task.

The general concept of how a DDPM model works can be seen in Fig 3. The general idea is that a model starts off with the ground truth image at step $x_0$ and with each step, a little bit of Gaussian noise is added. This process repeats until step $z$, where the entire image is Gaussian noise.

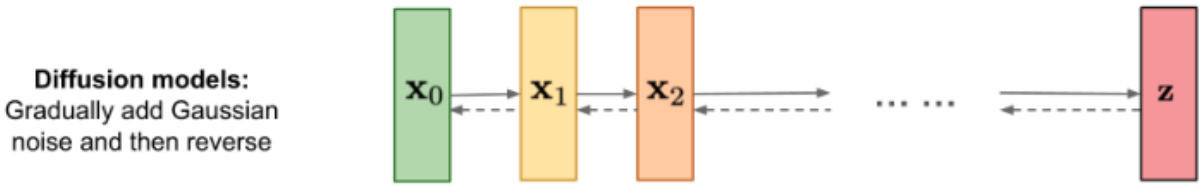


Figure 3: Outline of DDPM process [43]

At each step, the goal of the model is to learn how to restore the image to the step before it. Thus, creating a Markov Chain that is able to remove gradually remove noise from an image, until a clean image is produced. Each step of this Markov chain is responsible for a certain stage of how noisy the image is. Once training has fininshed, the model starts off with an image of pure Gaussian noise, as seen in Fig 4. The noise is then steadily removed using until a clean image is produced.

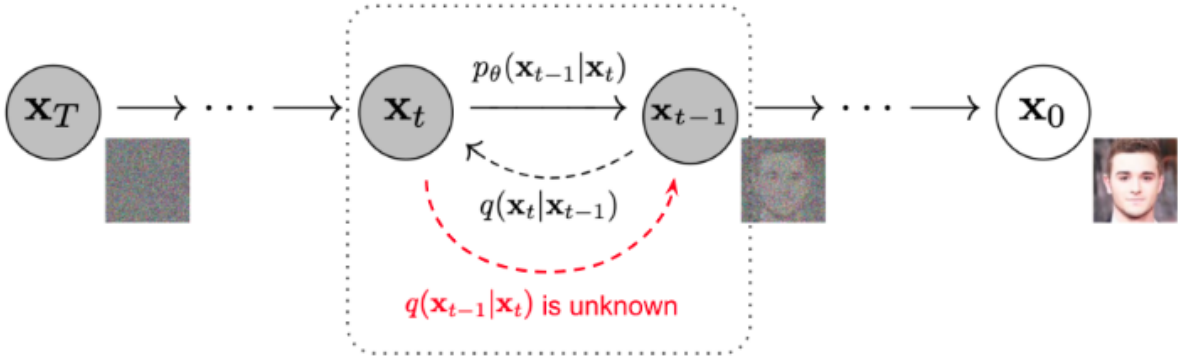


Figure 4: Example of Removing Gaussian noise [14]

In the case of video generation, the previous frame can be used to condition the denoising process in order to guide it to generate the next frame [45]. It is also possible to use a 3D UNET structure [15], where the 3rd dimension would be time. In all of these cases, the models are computationally intensive.

For this thesis, not the entire frame is generated, just the x and y coordinates of the chosen landmarks. The process is the same as described above with the Gaussian noise being added to each of the coordinates, and the model learning how to remove it. A similar task can be seen in Figure 5

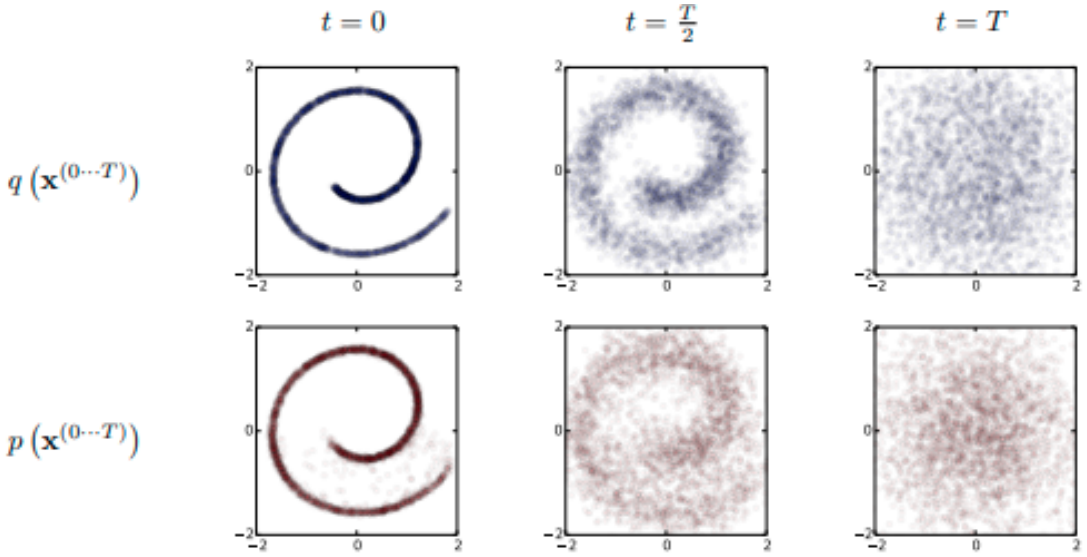


Figure 5: DDPM restoring coordinates [35]

Figure 5 shows how a DDPM model can restore the spiral pattern from pure Gaussian noise. The top row is how the spiral pattern changes as more Gaussian noise gets added to it. This would be the representation of what is happening during training. The second row is showing the model output at that point in the Markov chain. So, it is important to note that $t = 0$ for the second row is not referring to the input that the model is transforming, but instead what the model is predicting the image to be. As a result, each step in $t$ is the previous output of the Markov chain. So the initial values that the model will actually start off with when generating the values are the ones seen at $t = T$.

The last reason for choosing this type of model for the thesis is that it has been shown to be promising in handling these types of problems. At the time of starting the thesis, DDPM models are some of the best models for generating images, videos, and motion and landmark arrays. As a result, a DDPM model should be able to handle this type of problem fairly well, with the main issue being language translation from text to motion and not necessarily generating the information for each new frame.

## 1.4 Transformers

As mentioned, the DDPM was used to handle the generation of where the arrays of landmarks and motion. The Cross-Modality Linear Transformer was used to convert the sentence input into inputs that could then be used for the DDPM motion generator. This was done since transformers have recently started growing increasingly in popularity for models solving natural language tasks [10] [7] [5]. In this specific model, a Cross-Modality Linear Transformer is used.

Transformers have been developed for tasks that deal with sequential information. In these tasks, the previous inputs may be used at some undetermined point in the future. As a result, the model needs to be able to have a short and long-term memory of the inputs that it was given. A very good example of this is any task dealing with long text inputs. Since a word can be referring to another word in the same sentence, or earlier in the paragraph.

Previous to transformers, a common solution was recurrent neural networks(RNNs) [46]. The output of these types of models would be the prediction of the model, as well as information that would be used for the next input. This allows for the model to be able to make predictions based on the current input and any information that was previous to the current input. In theory the model would learn what information is important to keep and which can be discarded. In an example of a sentence, the previous information might be a noun that the current word is referring to. The issue arises in practice. RNNs have been shown to work very well for short-term memory, but they have poor long-term memory. This is due to a vanishing gradient problem [34].

In order to solve this a system called Long-Short Term Memory(LSTM) was developed [34]. The concept was that the model would be able to learn what was important to remember and what could be forgotten. In a basic RNN, the entire information that

is being passed onto the next state goes through a function. As a result, the model is not able to distinguish which information may be important and which can be forgotten. In an LSTM, the information passes through cell states, which always try to learn what information is important to remember and what can the forgotten.

Although there was an increase in performance using this type of method, the models were still shown to have long-term memory issues. It was shown that in long texts, the model would be likely to forget information that was presented near the beginning of the text when the end is reached. This is not surprising, since when thinking about most situations, a word at the beginning of the text will not have a large impact near the end. Another issue with RNNs is that the inputs have to be sequential, and as a result, an analysis of the words in the sentence will not be done in parallel [20].

The original transformer solves these issues by using Convolutional Neural Networks(CNNs) and Attention [17] [20]. The CNN element of a transformer allows for the text to be analyzed in parallel and helps reduce the distance between the output and the input words it could be using. That being said, CNNs alone were not able to solve how words depend on each other. This part is handled by the Attention, which focuses on specific subsections of the texts and creates hidden states that will be used to make the predictions, instead of just the word itself.

As mentioned, the thesis is using a cross-modality linear transformer. A cross-modality transformer is one that goes from one type of information to another. In this case, the model is going from text to either landmark positions or the motions of the landmarks. The linear part of the name refers to the time complexity in order to calculate the attention. The time complexity is linear making the process a lot faster. For these reasons, the cross-modality linear transformer was used in this model.

# 2 State of the Art

There have been multiple other papers [15] [13] [40] going over video generation, and more specifically papers on generating sign language videos from given texts [37] [26] [18] [36] [32] [31] [30] [21]. Video generation is a very broad topic and covers a lot of different specific cases. Many of which are not related to this thesis. It is still important to look at what has been done in those areas and see how it can be applied to much more niche topics such as this one. Even though this paper is about motion and landmark generation, many of the video generation models can be used for both. In this section, some of the solutions that were previously developed will be discussed, as well as a brief description of how they work.

## 2.1 Video Generation

The ability to generate videos has been one of the general problems that AI has been trying to solve. This has many different applications, from trying to predict a video, to add more frames to a video, to creating videos based on text inputs. A common solution used for video generation is Generative Adversarial Networks(GAN) [11]. These networks are very commonly used for problems that deal with images, so it is natural to extend them to videos as well. An outline of the structure can be seen in Figure 6.

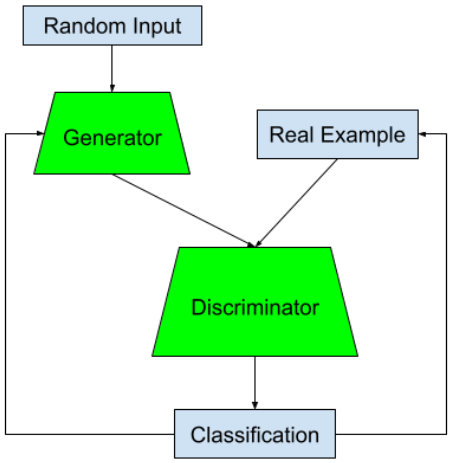


Figure 6: Outline of GAN Model

As seen in Figure 6, the basic concept behind a GAN network is that two networks are being trained to compete against each other. The first network is called a generator. This network is given a vector filled with random values. Its task is to generate an image that the discriminator model is not able to tell apart from a real image from this input. The generator generally is some sort of U-NET or Resnet structure and is typically more complex than the discriminator. An issue that can arise when training GAN networks is that one of the models might out-pace the other in learning the task and becoming way better than the other model. If this happens, then it might be possible that the model that is performing worse never learns how to do the task correctly, especially if this happens early in the training process.

There were other structures that were also used such as Variational Autoencoders(VAE) [44] [41] [21]. Generally, it was found that GAN networks produced better quality images [24]. Although GANs were better, VAE did provide some benefits, and there are models where VAE and GAN networks have been combined [9] [12]. Overall, the GAN network was still the more commonly used model for image and video generation.

It should also be mentioned that there are many other types of Diffusion Models that also are being used for image and video generation. One that recently has achieved a

lot of popularity is latent diffusion, specifically the stable diffusion model. These models function very similarly to DDPM, with the main difference being that latent diffusion models project the input into a smaller latent space. This is done by using variational autoencoders during the up-and-down sampling steps. This is an alternative method that reduces the computation overhead by leveraging lower dimensional spaces. These models have also been used to generate landmarks and motion values [4] [6].

Overall, the DDPM model is one of the newest and most prevalent methods for motion and landmark generation. There are other methods as well to generate these arrays, such as GAN. These models were shown to produce less realistic images than DDPM [25] [22] [24]. The goal of the thesis is to generate landmark and motion values, which this model has also been shown to be very capable of [42] [28] [47]. The model will be better described in Section 4.1.

## 2.2 Sign Language Video Generation

The previous solutions to sign language video generation follows very closely to trends in general tasks for video generation. This is unsurprising since if the model is shown to work well in a general area, it should work well for a niche area as well, as long as everything is adapted properly. As a result, this section will be discussing more of what was used in combination with the models used. Since most of the models were some variation of a GAN model. This will give an idea of what could be used in combination with this thesis or what to look for in future datasets.

The first method is at every frame, the joints extracted from the video are used to help create a prediction [18]. In this method, the model begins with being trained with the landmarks extracted from the video. These landmarks are typically points of interest, such as the joints in fingers, wrists, facial features, etc. During testing, the models use prerecorded landmark locations to help generate the image. Each frame that is generated in the beginning is an abstract frame, with the general idea of how a body is supposed to be positioned. The resulting frames are then passed through a style transfer to create a realistic-looking image, taking details from the original input image. Although not mentioned, an issue does appear since the first stage of generating the abstract frame is dependent on prerecorded landmark locations. It is possible that the model will have issues if the sign interpreters are not in the same position. This could have been fixed in multiple ways, such as cutting the frame to be centered around the landmarks or using the motion of the joints instead of their location.

Another method is to break the overall tasks into multiple smaller tasks. In this case, the first step is taking the text input and translating it to gloss. The gloss is then translated to a skeletal pose, which is the same as the landmark locations in the previous method. Lastly, the skeletal pose is translated into the image. The idea behind this is that by having each individual model focus on a specific task, they can excel at that individual task and provide better overall results. This is in comparison to having one model that

tries to do everything in one step. Although not mentioned in the paper [36], another advantage is that each of these models could be replaced by a newer model that has been shown to have a better performance. For example, instead of the RNN used in the paper, the text-to-gloss model can be replaced by a BERT model [7]. This replacement can be done independently of the models further down the line, as long as the output is in the correct format. Allowing for quick updates to the overall process. A disadvantage is that this method is likely to be slower than a model that does all of these steps by itself.

The last method uses a progressive transformer but mentions using several different augmentation methods [32]. The first augmentation is of future predictions, where the model predicts several future frames and not just the next one. This augmentation does not seem like it would be helpful for this thesis. Another augmentation that is mentioned is the addition of Gaussian noise to the frames. The Gaussian noise is applicable to most other models. In this case, the position of the landmarks can all have Gaussian noise added to them. Thus, slightly changing the positions or motions for training.

There have been clearly many different attempts at solving this problem. In several cases, it may be beneficial to take some of the ideas from these attempts and apply them to the process done in this thesis. The most helpful out of the ones listed will likely be the augmentation methods, that being said the augmentation methods need to be chosen carefully. For example, a mirror reflection would not be beneficial, since having different hands sign different glosses may change the meaning of the sentence. Overall, there are several ideas that could be attempted to be implemented to help improve the performance of the model.

## 2.3 Summary

This section has shown the previous works that have worked with video generation and translation from text to video of sign language. It has shown that the translation has closely followed the models that have been used for video generation. Previously these types of problems were solved with one of the multiple variations of a GAN model. As a result, most text-to-sign language models have also been GAN. The main takeaway from the previous reports for the text-to-sign language generation was ideas of how to handle the data and what to do with it. The use of a DDPM model for this type of translation of text inputs to sign language is a novel approach, so it is important to test it and try to see what can be used from previous solutions to help improve its performance. It is important to note that video generation was focused on in this section, is due to the papers researched in this thesis providing videos as the end result. These papers were still helpful due to many of them still generating landmarks as an in-between step between text and video for sign language.

# 3 Problem Statement

As discussed, the task that this thesis is focused on is generating a translation of a given text into an array of landmarks for certain points of interest. This array can be viewed as a video, in which each frame is a slice of the area. As a result, the thesis may sometimes refer to these arrays as videos instead of calling them arrays. This model would be helpful in multiple scenarios, ranging from helping to teach sign language, to possibly having it available in public areas for the use of translation. The first scenario could help people by being able to provide them with a representation of what a sentence may be signed. This is can be very helpful especially when the sentence is not common and the person wants to learn how to sign it. The second scenario could help increase accessibility for people that need to use sign language and may be a preferred method of communication compared to text.

The important parts of the problem are to generate the correct movements that seem natural and realistic. This means that the landmarks representing the points of certain body parts need to be properly distanced from each other. They also need to move fluently without sudden stutters or jumps. The landmarks can be later fed to a different model to produce a video of a sign language interpreter.

This thesis is mainly focused on proving that a DDPM model would be able to perform this task. Many of the previous works have mainly been with GAN networks and little research has been done into this specific area. The thesis will test several methods of preparing the data, such as if motion data is better than positional data. This again is a newer method, with most of the previous works preferring to train the models on the positional data.

For the thesis, there will be two main versions of the model. The first will have been trained on the data as originally provided in the datasets, with only filtering being applied. This will replicate the data format that is commonly used in previous papers [18] [37]. In a paper about using DDPM to generate human motion [47], a different approach was taken. In this case, the model was trying to predict the motion and positional vectors for the landmarks given a text. In order to keep the model simple, only the motion part will be experimented with. The comparison in results will give a better idea of which method is better, as well as the benefits and disadvantages of both.

Overall, the final goal of the thesis is to provide a good initial starting point for further research. The focus is more on providing the initial steps and providing research, than providing a final working example. Since a complete working model would require many more computational resources. The final product will be a better concept of how to proceed in this field, what could be tested in the future, as well as what may not be beneficial.

# 4 Architecture

Now that the basic concepts and the problem have been introduced, the procedure and models used can be described in greater in-depth. This section will describe how the model that is being used works. Then it will go into the LSFB dataset that was used. This will be an analysis of the dataset, a more in-depth description, and how it was filtered. Lastly, the training and performance will also be discussed. This will be how the training was set up and how and why the performance metrics were chosen and set up.

## 4.1 DDPM

As mentioned the model being used in this thesis will be a DDPM model. Specifically, the one which was created in the *MotionDiffuse: Text-Driven Human Motion Generation with Diffusion Model* paper [47]. This is due to it being a good starting point. Since it was created for solving a similar type of problem. The paper was also very recently published, which means that the model used will be a more novel approach to this task. Thus making it more beneficial to research it and how it handles this task. A deeper analysis of the model and its components will be provided here.

It is likely that the translation of the text to motion/landmark arrays representing sign language is more difficult than the original problem of creating the same arrays of text describing a motion. This is due to the outputs in the original task can be more general and open to interpretation. This is because the movements of the glosses have to be very specific, with one gloss being able to be mistaken for another if not done correctly. As a result, even if the model "understood" the text correctly and was trying to create the correct gloss, the human seeing the gloss could still misunderstand what was being signed. In the case of motion, this is not the case. Since walking can be represented in multiple ways, as long as the general motion is correct. Another reason is that the motions for movement are generally larger and more dynamic than for sign language. This allows for it to be easier for the model to not default to an average. As a result, this model may perform worse than in the original task.

The original paper [47] did implement several new methods to help improve the performance and calculations resources used by the model. The changes will be explained further in Sections 4.1.1, 4.1.2, 4.1.3, and 4.1.4. The processes and how the components interact with each other can be seen in Figure 7.

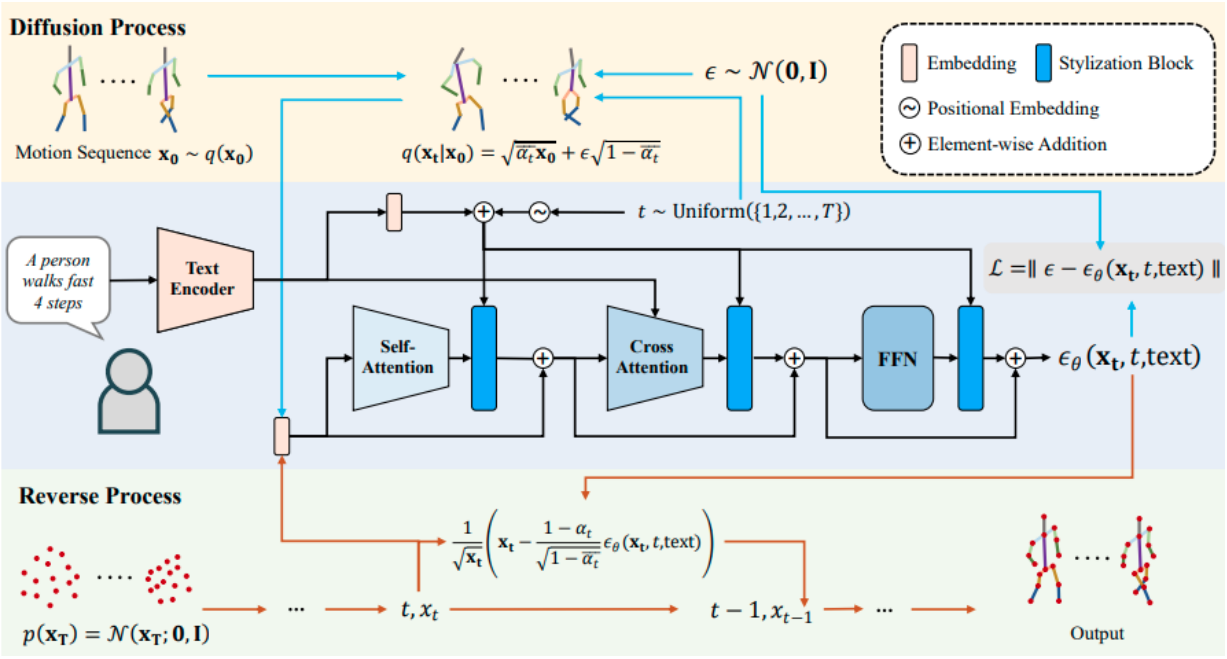


Figure 7: MotionDiffuse Model [47]: The blue arrows only happen during training, and the orange arrows only happen during inference.

As seen in Figure 7, there are two main phases that the model goes through. This is because, during training, the model is learning the steps to remove the noise. As a result, a different set of calculations is required. During training, the model gets a clean motion sequence and then steadily adds more noise to it. During the inference process, the model receives noise and follows the Markov chains that it has learned to remove the noise. This process will be better described in Section **??** about the diffusion process used in this thesis. It is important to note the $\epsilon_\theta(x_t, t, text)$ refers to the motion generator and neural network. The feed-forward network is not very specialized and will not be discussed in depth.

Another important point to notice is that generally, the denoising model utilizes a U-Net-like structure. In this case, the authors [47] decided to have it follow closer to a transformer-like structure. This is due to convolutional networks performing poorly when the input and output can be of variable length. It is important to mention that the model does require an input of how many frames it needs to produce. Previously, this would be addressed by having a set amount of frames being produced, with the extra frames either being blank or repeated. The model also has skip connections between multiple components, that help carry the previous features into the next stage, in case they were helpful further down the network. A U-Net-like structure is also not necessarily necessary, since not an entire image is being generated, but only the landmarks. As a result, a lot less data is being input, which is the main advantage of the U-Net structure that it is able

to compress the original data to a smaller dimension so that the neural network can be smaller and run faster. This section will be broken down further into the path that data would take through the model.

### 4.1.1 Text Encoder

The first part of the model that the data will encounter is clearly the text encoder. This part is responsible for encoding the text into a format that can be used by the rest of the model. The encoder is based off of the transfer developed in the paper by Vaswani [39]. This consists of an embedding layer, a multi-head attention module, and a feed-forward network.

The embedding layer is responsible for getting text features from the input sentence. These features can information such as the part of the speech that the word is functioning as or the root of the word. The original project had the other information extracted for specific words like certain body parts, descriptions, or actions. This would be very helpful since it allows for better information to be passed to the model. In the case of this thesis, it would not be applicable. It would also not be possible to apply lists like these, since the goal of translation is too general and a specific list could not be made.

The multi-head attention module gets the query feature vectors, key feature vectors, and value feature vectors. These terms come from the Vaswani paper [39] and are in relation to the attention function. The query vector and a set of key-value pairs are usually used to produce an output. This output is then passed to a feed-forward network.

There are also several other techniques used to help create the text encoder. At the end of the process, the text encoder will be outputting a vector based on the features extracted from the input text to the cross-attention component. This component will be better described in section 4.1.3.

### 4.1.2 Linear Self Attention

The other input to the cross-attention block is the output from the linear self-attention component. This component does not use text input and instead focuses on finding correlations between the time frames. These frames will either contain the motion values or position values, depending on the format the data is being trained on. In this thesis, both of these formats will be tested. It is important to note that in this section these arrays will often be referred to as videos and use similar terminology as when referring to videos.

As mentioned in section 1.4, the larger number of frames and elements creates a greater time complexity in order to do all the calculations. In order to try to decrease this time complexity as much as possible, an Efficient Attention architecture [33] was used. By using this type of attention, the time complexity was reduced from being quadratic down to being linear. This will allow for the model to create much longer arrays, as well as have more information per slice of the arrays. As a result, being able to produce better videos from the landmarks and motions that were generated.

The other reason the original paper [47] opted for using this type of attention model, was due to it being able to use global information. This is in contrast to the typical attention models which rely more on pair-wise relations, and thus are unable to gain as much information about semantic meaning for the frames being analyzed. This means that these types of networks are better suited for things such as video generation, in addition to having a faster time complexity. A comparison of the two types of attention architectures can be seen in Figure 8.

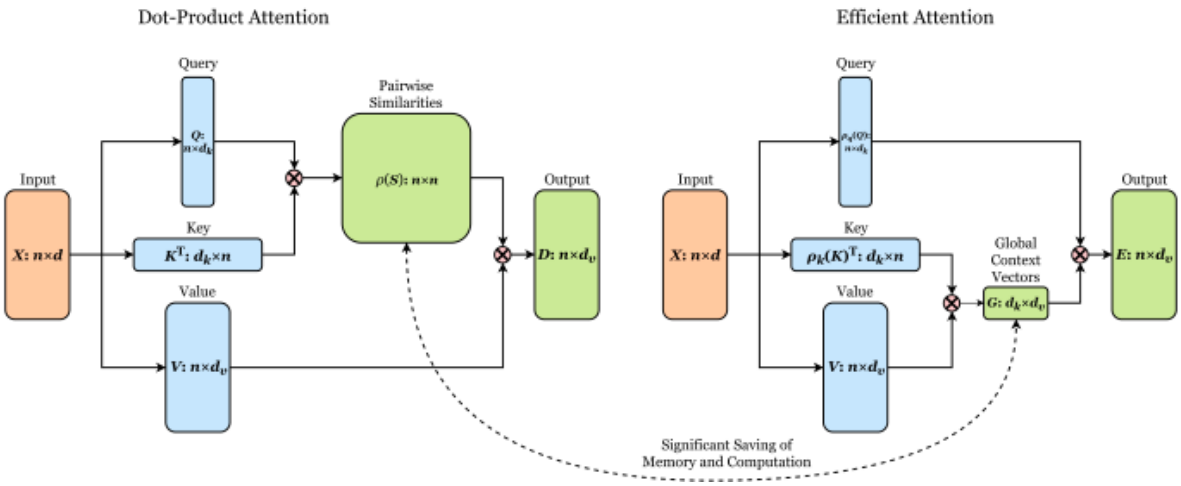


Figure 8: Linear(Efficient) attention vs Classical(Dot) Attention Architecture [33]

The Figure shows that the two architectures have very similar functions. The main difference besides the scaling normalization demonstrated by the $\rho$ is the order that the operations are being carried out from $(QK^T)V$ to $Q(K^TV)$. This small change is what changes the time complexity and also what changes the model from using pairwise similarities to a more global correlation.

These improvements are going to help the model better learn how to do its future predictions based on the overall video. It will also significantly speed up the training process. The memory required to run the model will also be significantly decreased, compared to a classic attention structure. The features extracted for the entire video will be passed to the cross-attention component as seen in Figure 7.

### 4.1.3 Cross Attention

As mentioned in Sections 4.1.3 and 4.1.1, the text encoder and linear self-attention components will be feeding input features to this component. As a result, this component is responsible for learning how to combine the motion/positional features with the text features. This module will also use the Efficient Attention architecture, in order to speed the process up and save memory.

Since the structure does not change much, the overall process between the linear attention component and the cross-attention process is very similar. The comparison between the two architectures can be seen in Figure 9

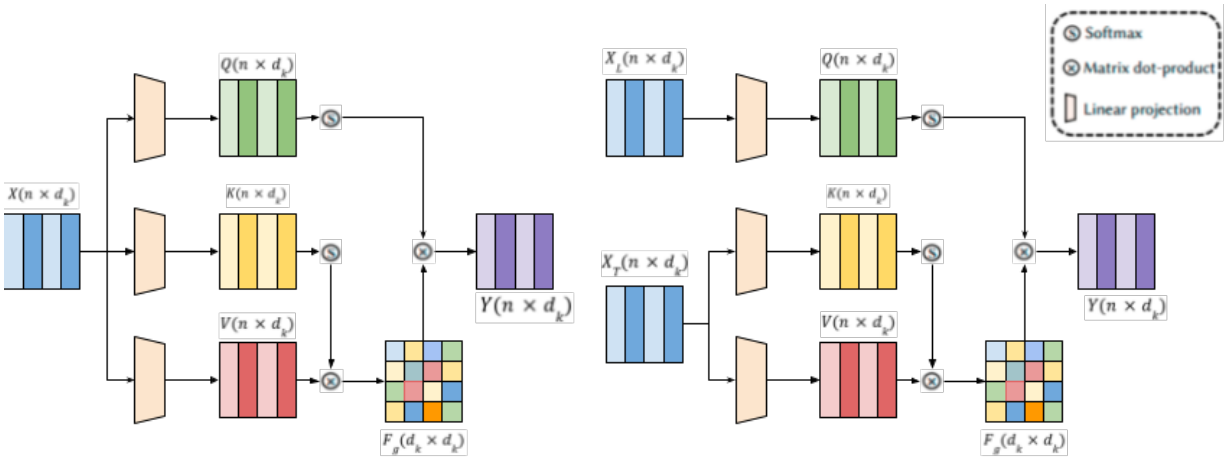


Figure 9: Comparison between Linear Attention(left) and Cross Attention Components(right). [47]

As Figure 9 shows the main difference between the two architectures is how the inputs are used. Clearly, the cross-attention component needs to be able to handle two inputs, instead of just one. As a result, the input from the text encoder is used to obtain the key and value vectors, and the linear self-attention input is used to get the query vector.

The split for the inputs between the vectors happens for the following reasons. The attention model is mapping the query vector($Q$) and a set of key-value pairs($K$ and $V$) to an output [39]. This means that one input has to be assigned to the query vector and the other input vector to the key and value vectors. It is likely that the text input gets assigned to the key and value vectors due to there being the possibility of more features from it being used. Since the text is more important and likely more complex than the motion/landmarks, it is important to use more of its features than the features obtained from the motion/landmarks.

Overall, the cross-attention component functions very similarly to the previous linear attention component. There were mainly changes due to there being two inputs into the component. This is not surprising since the Efficient Attention architecture is likely the best-suited attention architecture for tasks such as these, since it lowers the training time, and the memory required and focuses more on the global correlations. The output of the cross-attention component is then fed to the feed-forward network.

#### 4.1.4 Stylization Blocks

The stylization blocks appear between each of the components and the next step, as seen in Figure 7. The stylization blocks take in the features produced by the text encoder, as well as which time stamp the diffusion process is at. This was done to help reinforce the text motion, instead of possibly learning the average motion/position overall. The output of the stylization blocks is determined based on the following equations:

$$\begin{aligned} B &= \psi_b(\phi(e)) \\ W &= \psi_w(\phi(e)) \\ Y' &= Y \odot W + B \end{aligned} \tag{1}$$

In the equations above, e stands for the summation text embeddings provided by the text encoder and the time stamp, $\psi$s and $\phi$ are different linear projects, and $Y$ is the output of the previous output. These stylization blocks will provide a stronger bias to the text input into the output being produced.

#### 4.1.5 Summary

The MotionDiffuse model [47] is clearly well suited for this type of problem. Through the use of the text encoder and multiple stylization blocks, the model attempts to enforce the text features to dictate the output. This is due to the text features being added after every component before the motion generator and the neural network at the end. The model did end up including some unique features to help with the performance of this task.

The main unique features of the model come in the attention components. In both cases, the model is using a new type of self-attention called, Efficient Attention. This attention is similar to the general attention architecture, but with a couple of small changes, it is able to speed up the computational time and use less memory used during computations. It also helps shift the focus to global correlations, instead of pairwise relations.

As mentioned the model also has multiple stylization blocks between each component. These blocks repeatedly add the same text features from the text encoder to the input of feature stages. The style blocks themselves are not a novel concept, but the addition of them in the model is new compared to some of the other DDPM models.

Overall this model has clearly several new implementations that help it be extraordinarily useful for the generation of motion and landmark arrays from text inputs. It is able to have a faster training time and use less memory than other similar models due to the Efficient Attention architecture. It also has a large emphasis on the text features extracted from the text encoder. These both will likely be vital to help the diffusion model learn how to translate the text input into the correct gloss motions.

## 4.2 Dataset

The dataset being used in this thesis is the French Belgian Sign Language(LSFB) dataset [8]. This dataset was chosen due to how large it was and that it included the landmarks of the signers. The overall dataset has 100 signers and over 85000 samples. It is important to note that each gloss that happens in the video is considered to be a separate sample. This is by far one of the larger datasets available publicly available, as demonstrated in Table 1.

Table 1: Dataset Information

| Dataset | Classes | Subjects | Samples | Language level | Annotations |
|---|---|---|---|---|---|
| LSFB Cont. [8] | 6883 | 100 | 8500+ | Word/Sentence | landmarks, end/start, french translation |
| RWTH-PHOENIX-Weather [19] | 1200 | 9 | 1200 | Sentence | Sentence start and end, landmarks |
| MS-ASL [38] | 1000 | 222 | 25513 | Word | Word start and end |

It is important to note that even though the MS-ASL dataset states that there are 25,513 samples, many of these are not reachable any longer. This is due to the MS-ASL dataset being dependent on YouTube videos. Since the dataset has been created, many of these videos have either been deleted or are no longer public. As a result, the number of samples has decreased significantly. It also does not provide the landmarks of the signers, which is important data to have for this thesis. As a result, this dataset could not be used, since using models that would provide the landmarks and provide uncertainty if the tracking was done correctly.

Another disadvantage of the MS-ASL dataset, since it is created from a collection of YouTube videos, the format is not standardized. The signer does not appear in the same part of the frame consistently between videos. The positioning of the signer may be different, some were standing and others were sitting down. Since there were so many variations between the videos would make the problem too difficult for the model to learn correctly.

The most common dataset used in other papers is the RWTH-PHOENIX-Weather(RWTH) dataset. This is likely due to it being one of the largest and most consistent datasets available. The dataset is made from videos of signers signing the weather reports on a weather channel. As a result, the camera angle and positioning of the subject will be relatively similar in all of the videos. The landmarks are also provided with the dataset, which was often used to help the other models generate the videos.

The main reason that LSFB was chosen is due to it having a much larger sample size. The larger amount of signers is not as important. This is due to currently the model is only producing the landmarks that are moving and not the video of an actual person. As a result, the variance that the different signers bring is the speed that they sign and the distance between the landmarks being different. The signers in this dataset have two main camera angles and are generally positioned in the same area of the frame, although it is not as consistent as in the RWTH dataset, as seen in Figure10.

Figure 10: Example of signers

The figure also demonstrates that in some videos, the second speaker is sometimes seen. This is demonstrated in the leftmost frame. This will not affect the data being trained on, since they will not show up in landmark data. This also demonstrates how the dataset was created. The dataset is a collection of videos of two people talking to each other in sign language. As a result, the dataset becomes very varied with the glosses that it is using. This could be a disadvantage since the models will have less opportunity on training on the same glosses.

A piece of information that the model required that was not provided by the LSFB dataset, was the part of speech and lemma of the words in the sentence. These were obtained by using the Standford NLP CoreNLP [23]. The specific version for French had to be found and used. This CoreNLP provided the part of speech tags and lemmas for the words.

### 4.2.1 Filtering

There were several reasons why it was necessary to filter the dataset. The main reason was that some of the videos were too long. In the paper [47], the model was created to make videos of only 240 frames long. Since the videos are shot at 50 frames per second, it means that the length that would be allowed using this restriction would be about 4.8 seconds. Since this limited the number of sentences available too severely, it was doubled to 500 frames. This will greatly complicate the model, but to counteract the increase in length, the output vector was significantly reduced.

As mentioned, the output landmarks were limited in order to try to reduce the complexity of the model. The original dataset provided 23 landmarks that were being tracked in the pose annotations. These annotations were the more general body landmarks, such as shoulder and facial features. The dataset provided 42 landmarks being tracked in the hand's annotations. These were more specific to each joint in the hand, with there being some overlap with the pose annotations. The combination of the two annotations was reduced to 21 landmarks. The landmarks were the tip of the fingers, the wrists, elbows, shoulders, corners of the mouth, and ears. Although it would not be possible to create a video of an understandable gloss using these landmarks, it would be a good initial starting

point.

There was also filtering applied when the dataset was being split into the test and validation sets. Since the main objective was to see how well it would translate the text to a motion, it was decided that the test and validation sets would only contain words that the model would have previously seen. The words in the validation and test datasets would appear at least ten times. The training dataset did not have this limitation. This is due to the model being able to learn how to transition between two different positions more naturally by the inclusion of words that it will see. This is due to it seeing how motion is applied from different positions of the landmarks.

In the end, there ended up being 3081 samples. Generally, a dataset is split .8 train, .1 test, and .1 validation. In this case, since the test and validation were dictated by how many times a word is seen, the split ended up being closer to .88 train, .05 test, and .07 validation. This means that there are about 151 test samples, 190 validation samples, and 2774 training samples. An effect of having such small datasets may be that the model selection, during training, became less accurate. In order to compensate for this, it was decided to have a larger validation dataset than a test dataset.

### 4.2.2 Analysis

This section will provide an analysis of the overall dataset, as well as the datasets created. The main focus will be the distribution of words since this is what will be focused on by the models. The analysis of glosses will help provide a better understanding of sign language, as well as understand see what motions are repeated.

The first part of the analysis will be done on the overall datasets. This will include the words and glosses that happen in the entire dataset. In Figure 11, the histograms of the most common words can be seen, and a more detailed result of the top five most common words can be seen in Table 2.

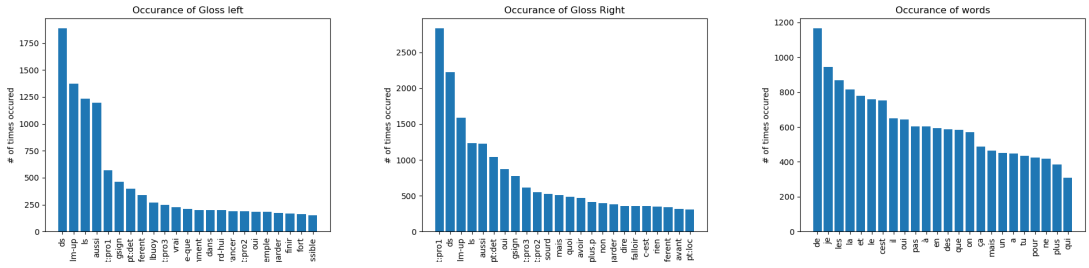


Figure 11: Breakdown of gloss and word occurrence for overall dataset

Table 2: Most frequent Word and Gloss Occurrences

| Word | Occurance |
|---|---|
| de | 1166 |
| je | 946 |
| les | 868 |
| la | 814 |
| et | 780 |
| total | 42361 |

| Gloss Right | Occurance |
|---|---|
| PT:PRO1 | 2835 |
| DS | 2223 |
| PALM-UP | 1587 |
| LS | 1232 |
| AUSSI | 1226 |
| total | 52666 |

| Gloss Left | Occurance |
|---|---|
| DS | 1886 |
| PALM-UP | 1373 |
| LS | 1236 |
| AUSSI | 1198 |
| PT:PRO1 | 569 |
| total | 30432 |

As both Figure 11 and Table 2 show, the most common words are fairly basic and are some sort of prepositions, articles, pronouns, or conjunctions. This is not surprising, since these would be used in most sentences, while specific nouns and verbs would appear less commonly since they depend more heavily on the situation that is being talked about.

A more interesting detail can be seen in the gloss tables in Table 2. It can be easily seen that the right hand is much more often used to sign, almost twice as much as the left. The most common glosses for these hands are also the same but appear in different orders. This large discrepancy may lead to the model performing worse with the left hand than the right since there is less amount of data for it.

In order to see how the split affected the distribution of words between the datasets, the breakdown of the words was also observed in each dataset. These distributions can be seen in Figures 12 and Table 3. Since the model does not train on glosses, these were not provided in the individual datasets.

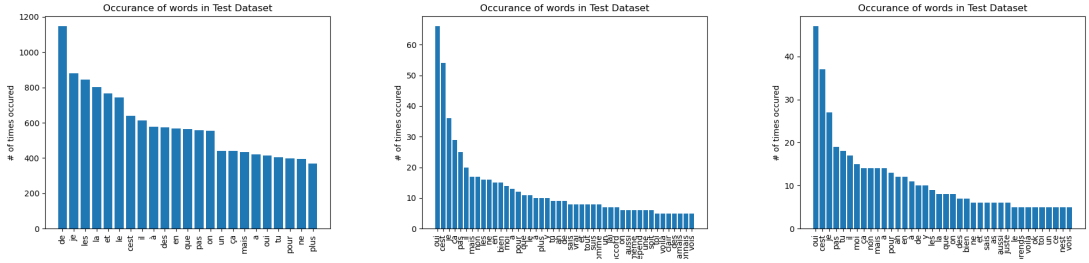


Figure 12: Breakdown of gloss and word occurrence for individual datasets

Table 3: Most frequent Word Occurrences in individual Data sets

| Training | |
|---|---|
| Word | Occurance |
| de | 1147 |
| je | 882 |
| les | 843 |
| la | 803 |
| et | 766 |
| total | 40471 |

| Validation | |
|---|---|
| Word | Occurance |
| oui | 66 |
| cest | 54 |
| je | 36 |
| ça | 29 |
| pas | 25 |
| total | 928 |

| Test | |
|---|---|
| Word | Occurance |
| oui | 47 |
| cest | 37 |
| je | 27 |
| pas | 19 |
| tu | 18 |
| total | 732 |

The Table 3 and Figure 12 show that unsurprisingly the training dataset remains to be fairly similar to the overall dataset. This is because of the high percentage of the overall dataset that the training dataset consists of. As a result, if this were to differ it would implicate that something went wrong when the datasets were split.

The validation and test datasets due differ from the distribution of the entire dataset. This is clearly seen in the validation and test tables in Table 3. Almost all of the most common words in this table are new when compared to the table for the entire dataset. The figures for these datasets seen in Figure 12 also show how often certain words have occurred changed. Although it is important to note that most of the words have remained to be the same. It is not clear why this would be the case, but it may be due to that some words are more common in short generic sentences. These types of sentences would be the ones that would meet the requirements that every word in the sentence has to show up a certain amount of times in the dataset.

Overall, the analysis of the datasets has shown that the split has worked as intended. The train and test datasets do have a different distribution of words, but this is likely due to them having more generic sentences. The analysis has also shown that the right hand is more commonly used when signing, which may lead to a worse performance when producing signs with the left hand.

### 4.2.3 Summary

The dataset chosen for this thesis was one that has not been commonly used, which is likely due to it being a newer dataset. It was chosen for its size and that it provided landmarks for the videos. The dataset also required a lot of filtering due to the limitations of the resources and model, as well as the desired requirements for testing. The filtering has shown to have had an effect on the distribution of the words present in the dataset, but this likely won't have a large effect and most of the most common words have remained the same.

The two other large datasets found called RWTH-PHOENIX-Weather [19] and the MS-ASL [38] were both under consideration of being used. They were not used due to either being smaller than the LSFB dataset, not providing landmarks, or not being as

standardized. The LSFB data set has been created in 2021, which is the likely reason why it has not been used more often in sign language papers.

The filter done on the dataset was making sure that all of the video being trained on would be under a certain amount of frames. This was done to limit the complexity of the problem that the model would be solving. The number of joints being output was also limited to try to compensate for the larger amount of frames being produced. Lastly, the test and validation dataset were purposely made to have words that were trained on, but not repeating sentences in order to test the model on creating the signs it knows and not trying to guess at signs it does not know.

The analysis has shown that the splitting of the datasets has changed how often certain words occurred. Although that being said, the top most common words mostly remained the same. It is unlikely that this will have a strong effect on the model's performance. The most common words were unsurprisingly things such as articles, pronouns, and prepositions, due to them not being specific to the topic of conversation as verbs and nouns are.

Overall, the dataset is best suited for this thesis due to its large size, providing landmarks, and entire sentences. The filtering applied to the dataset was done to help reduce the number of resources required to train the model and more accurately reflect the goal of training the model. The filtering did affect the distribution of words in the dataset, but it is unlikely to have a strong effect on the dataset. These decisions all are likely to help improve the model's performance and provide results that are more representative of the goal of the thesis.

## 4.3 Training

The training for this model followed standard procedure. It was done in Google Colab and Kaggle. This means that the model was using a K80 GPU when being done in Colab and for Kaggle it was using a GPU P100. In the case of the Colab model, it was necessary to save the model and then continue training. This likely did not have an effect on the performance of the models produced. In table 4, the settings for the model can be seen.

Table 4: Settings Configurations

| Setting | Value |
|---|---|
| Times | 1 |
| Epochs | 200 |
| Batch Size | 2 |
| Sampler | Uniform |
| Workers per GPU | 4 |
| Loss Type | MSE/Weighted MSE |

The *Times* setting is how many times the dataset was duplicated. This was done previously in the code due to how long it took to initialize multiple data loaders for the workers. This was resolved in later versions of the code, but this setting was not yet removed. As a result, in order to keep track of how many times the model goes through the dataset, it was set to one.

As mentioned before there were two loss types tested in the model. The first one is a general mean square error loss(MSE). This was used for both the positional and motion data formats. In this format, all of the joints are weighted equally and the loss is equal to the average squared error between them. The weighted MSE was tested to see if the model could be forced to focus on certain joints more specifically. Since the hands are the most important there was a higher weight put onto the errors in those landmarks than the rest of the landmarks, also since they are the landmarks that will have the most amount of change.

The sampler setting is what the sampler method is used in the diffusion model. There are many different types such as k_euler, k_huen, plms. The sampler is used by the model during the denoising process. In future work, it would be beneficial to try different types of samplers, but it was not done in this thesis.

It is important to note that not all of the settings are directly accessible. The very basic settings such as times, epochs, or batch size could be easily changed when the script was executed. Some other settings such as Loss Type, required going into the code and either manually adjusting them or creating the code for it. Overall, the training for the model in the thesis was fairly simple and straightforward.

## 4.4 Performance

There are several different ways that papers that focused on sign language generation have tested their models. These typically were one of the following: human analysis, sign language model interpretation, and comparison to the original video. The method chosen for this thesis however was a mean square error metric across the entire video produced from the motion/landmarks. The thesis will also use the MSE score that is achieved for each movement of the landmarks between frames. This section will further explain the performance measurements, which are commonly used, and the reasons for not using them.

The method that provides the best information on performance is the human analysis method. The human analysis and sign language model interpretations are very similar to each other. Human analysis is considered to be the best way to measure the performance of the model. This is because it is most similar to what the model will actually be used for. If the human is able to interpret the output correctly, then the model is clearly working. A human can also give feedback if there is something incorrect, such as speed or appearance that the model analysis would not be able to provide. The main disadvantage of this method is that it is slower and more labor-intensive than any of the other analysis methods used.

This type of measurement was not chosen for several reasons. The main one is that due to the small number of landmarks and displaying the data in a skeletal format, it would likely be impossible to correctly interpret the model's output. The other big reason is that it would not be possible to find enough interpreters to give an accurate analysis of the model. As a result, this method would not be possible to implement.

The sign language model interpretation is another common method of doing an analysis of the results. It is good since it is possible to quickly see the accuracy of a lot of generated videos. This requires another model that was trained to interpret sign language videos to text. It is likely if this type of model is able to correctly translate the produced videos, then a human would be able to understand it as well. The main disadvantage is that even though the model may be able to translate it correctly, a human may not be able to or will dislike something about the video generated, thus not wanting to use the trained model.

This is another method that would have been good to use. Although for similar reasons it would not be possible for the thesis. This is due to the low amount of landmarks, any model that used landmarks to create the translations would have to be retrained to function on the lower amount. It is likely that due to the small amount even a model would not be able to accurately interpret the videos. This is due to all of the joints in the fingers missing in the created datasets. As a result, any bending in the fingers will not be seen and could instead be interpreted that the angle of the hand is changing. The other issue is that many of the models that would be readily available most likely were trained on an English dataset or a German one. So even if there were enough landmarks, the model would have to be retrained for this method to work.

The last method of comparing to the original video is used when the paper is trying to create a video of a person signing. This is a general method used in a lot of video or image generation methods. Since if the generated video is close to the original video, then the signing should be close enough to be easily interpreted. This is generally done with measurements such as Chamfer Similarity(CS), Structural Similarity Index Measure(SSIM), and Peak Signal-to-Noise Ration(PSNR). This type of method is very advantageous since it is the simplest to set up and doesn't require many other resources outside of the model being created. The disadvantages are also that it may be possible that the model is scoring highly in the similarity, but a human would not be able to understand what is being signed, or would not want to use the output of the model. The reason why this method was not used was that a video is not being produced for this thesis. The produced result from the model would be a set landmark, and as a result, these methods would not make as much sense to check how well the performance of the model is.

In the end, the performance metric used for the thesis will be the mean square error(MSE) score. There will be two main analyses done using this score. The first is measuring the predicted motion compared to the ground truth motion. This type of measurement is the one that makes the most amount of sense for a problem like this. In the case where the outputs are landmarks, instead of predicted marks, the landmarks will be converted to motion first, and then the performance will be calculated. The reason for

this is that if the motion is converted back to landmarks, then the errors from the very beginning will compound until the end. Thus, giving a higher artificial error to the motion models than to the landmarks model. This type of measurement will provide a good idea of how well the model is doing when predicting the transitions from frame to frame.

The other method will be seeing how the analysis would have performed if the motions present in the predictions were continuously added to the predicted frames. This will provide an idea of how the error of the model over the entire produced video. While this method provides an idea of how well the model performs the entire video. This is useful, since if there is too much error aggregation throughout the video. Then the video produced will not be useful.

The two methods used produce a better way of comparing the two models. Sine the values used in the motion dataset are significantly smaller than the positional dataset. As a result, if the motion model was only compared to the MSE measurements of their domain. The positional model would always appear to be performing significantly worse. By converting the two models into both domains, it is possible to do a more fair comparison between the two models.

The frames from the produced videos will not be analyzed deeply. This is due to them not providing useful information due to the frames being still and not being able to see the motion. A brief description of what happened in the models will be provided. The comparison will mainly be between the landmarks model and the motion model since it will be difficult to make any conclusions when comparing the weighted vs non-weighted models.

There are clearly many different types of ways to measure the results of the model for this type of problem. These methods would have to be applied when this model has been further developed since currently, the methods would not produce an accurate assessment of the performance. As a result, a much simpler method MSE method was used for this thesis, in order to obtain results that are more accurate to the performance and would function correctly. In order to compensate for the simplicity of the measurement, multiple methods were developed to get a better idea of how each type of model performs.

## 4.5 Summary

In summary, this thesis has several new components in it. The model used in the thesis has only recently been developed, as well as the dataset that it was trained on. The metric results used had to be simplified due to how simple the output is, and the outputs would be incompatible with the more sophisticated measuring techniques.

The model being used is a DDPM model that was specifically designed for generating a video of motion from text inputs. The authors of the model [47] have adjusted the model in order for it to be able to function better for this purpose, as well as implementing new features, such as the Efficient Attention architecture, to help speed up training and have it use fewer resources. As a result, the model would have an easier time generating the

arrays of motion and landmarks.

As mentioned the LSFB [8] dataset is also fairly new and has not been used often previously. Due to its size and provide features, it is a better option than the two previously commonly used datasets of RWTH-PHOENIX-Weather [19] and MS-ASL [38]. There was filtering applied to the dataset in order to help narrow how many frames are being produced, as well as splitting the datasets such that the test and validation dataset contained words previously trained on. This was shown to have an effect on the distribution of words but likely did not have an effect on the results of the overall model.

The performance metric used for this thesis is a very simple one, but it is due to none of the more complex metrics being a proper way to ascertain to performance of the model. Since there were so few landmarks being produced, it is unlikely that any human or another model would be able to identify what the gloss is supposed to be. Since also only the landmarks or motion vectors are being produced, it does not make sense to use a metric that is generally used for videos, which would be a common approach for measuring the performance. As a result, a simple measurement of the error was decided upon to get an accurate understanding of the model's performance.

This concludes the architecture section of the thesis. The new DDPM model clearly benefits from the new attention structure and use of transformers and stylization blocks. Although the new dataset makes it more difficult to compare to similar papers, it provides the benefit of a much larger dataset with the necessary information for the thesis. The mean square error performance metric, although very simple, is the best suited for the thesis, due to the simple output of the model.

# 5 Results

As mentioned before the results will be broken down into two main comparisons. This will be the motion models trained and the positional models trained. This is due to them having very large differences in how the results for the models appear. For example, in the case of the motion models, most of the loss measurements appear to be much smaller, due to generally the motion model working with smaller values. In order to limit comparisons between the models, they were divided in such a fashion. A common issue in both models was that the models learned the average motion apparent in the sentence.

## 5.1 Motion Models

The first model being discussed is the motion model. Between the two types of models, the motion model did appear to be more stable, with less noise in the loss and sudden major decreases in performance. The model was not able to learn the motion of the gestures and instead learns the average motion. As a result, when the produced landmarks created using the motion vectors are converted to video, the models are not moving in relation to each

other.

The motion models loss implicates, shows that the DDPM models used tend to learn very quickly. As shown in Figure 13, the models had a very fast learning rate and then slowed down dramatically. The loss in the weighted plots, seen in Figure 13, is likely higher due to the fact that the loss is being multiplied. As a result, the model has to do more work to have the same score as the unweighted model. It is important to note that an averaging filter was applied to both plots, in order to get a better idea of the trends in the model.

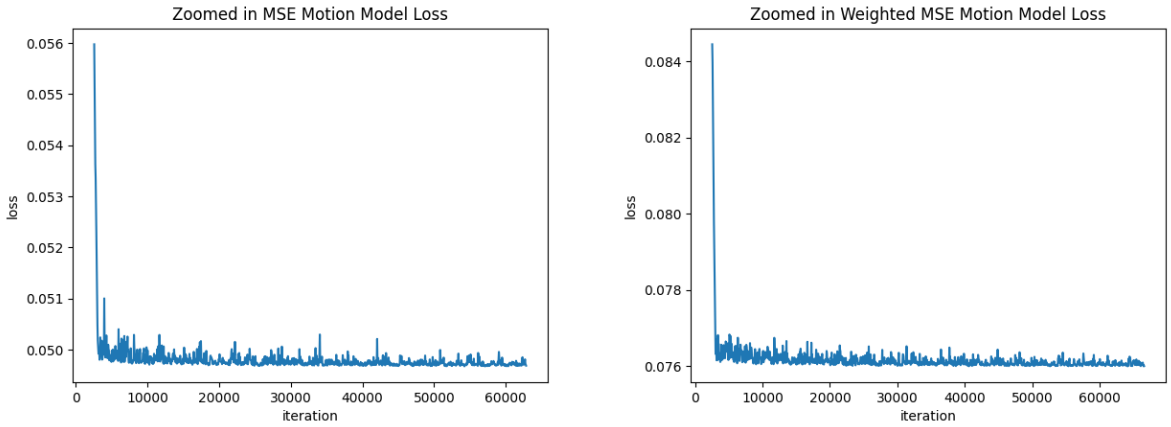


Figure 13: Zoomed in Loss of Motion Models

In Figure 13, it is seen that when the model finished training it seems as if it was near its maximum performance. It is difficult to tell if the model has actually reached peak performance, or if it is becoming more stable. As a result, the noise is decreasing and making it appear to have not reached the optimal state. Previous tests have been done and it had not improved significantly after this point. As a result, these results will likely be near the results that would have been achieved if these models were trained longer. Although the poor motion results may show that it would have been better to continue training and it was just learning very slowly.

As mentioned in previous sections, there were two procedures used to measure the model's performance. In the case of motion, these results have provided very interesting results, with a clear preference as to which model performs better. These results can be seen in Table 5.

Table 5: Mean Square Error Results for Motion Models

| Model | Landmarks | Motion |
|---|---|---|
| Motion Weighted MSE | 940.226 | 0.392 |
| Motion MSE | 0.010 | 9.868e-05 |

The initial results seen for the landmarks procedure in Table 5 are at first surprising. As discussed previously the reason for this is that the values that the motion vectors use are significantly smaller. As a result, a direct comparison makes it appear as if it is learning the motion a lot better than the landmarks. The compounding error mentioned previously is likely not having a significant impact. Since the motion from the MSE loss model tends to be zero, producing a still frame. An average motion is still being applied to the landmarks, but it is not equivalent to all of them. This resulted in a video where the landmarks would all be sliding in on direction, and slowly getting further apart. This is likely due to weights changing what the model sees as the average motion. Since the motion is constant as it was in the non-weighted MSE loss model.

Considering this result, it is not surprising that the regular MSE loss model performed much better compared to the weighted one. Although it is interesting to note that in the positional model, this is not observed. In Table 8, seen in the Appendix Section 9.1, it does not appear that the model performed worse due to it prioritizing learning certain landmarks over others. Since the ratio for the higher valued landmarks, such as fingers and hands, is lower than what it ended up being for the normal model. The model being weighted also does not have an effect on the videos produced.

The results seen in 8, also show that the model had the most error when predicting the motions of the hands and fingertips. This is not surprising, since these are the landmarks that will be moving the most. The rest of the landmarks are likely to remain relatively still. As a result, the predictions for their motions are not as diverse and complex as for the fingers. This would have been expected even if the motion model was producing significant motion, and not just the average motion.

An interesting result can be seen in Table 10 seen in the Appendix Section 9.4. It would be expected that a longer sentence input would produce a higher overall error due to it being more complex. This is not the case due to there being longer sentences that had a high accuracy, and simple sentences that had a low one. They appeared evenly distributed throughout the performance range. Only in the very top accuracies did it appear that shorter sentences were preferred.

This is likely again due to the motion model only producing motion vectors of values close to zero, causing the landmarks to stay still. As a result, the measurements shown in that table are how much the sentences are different from the average, and not how well the model is actually predicting the motions. This would mean that the sentence *moi oui* has a large array of motions that differ from the average of a sentence like *ah oui*. The

two sentences are going to have a similar length, but *moi oui* had either faster or larger motions than the *ah oui* which is what caused the issues.

Although it seems that the motion model overall was able to produce a fairly good performance when compared to the positional model. This is not accurate, since the model was only better at predicting the average motion. As will be discussed further in this thesis, this is likely due to the MSE loss function and it's preferring to minimize the error by predicting the average, instead of increasing the error by trying to predict the actual motions of the gestures.

## 5.2 Positional Models

The results for the positional model closely follow the observations made with the motion models. This is not very surprising since the model has not changed between the two. Only the format of the data that the models are learning has changed. As a result, any issues that were experienced with the motion are likely to translate when learning the positions. The positional model also did have an issue that it would produce the landmarks in what seemed to be an average position for the text. The main difference was that the landmark did have more motion present in them. Although it is difficult to tell when this motion is random noise and when it is the model trying to replicate a motion.

A major difference, as mentioned in the previous section, is that the positional model was less stable during training. This can be seen in Figure 14. The large spike near the end of training happens in both the weighted MSE and normal MSE loss. It was also something that happened during multiple tests, as a result, this was not a unique experience to this run.

As mentioned before, an issue with the positional model is that the outputs would often not change from the initial frame that it predicted. This position is likely the average position for the sentence. It is possible that this spike occurred when the model tried having the landmarks have a larger motion from the average position. The error from the wrong motion could have been significantly worse than staying with the average position, which caused the model to return back to the average position, and the loss results. This spike also makes it hard to make a good comparison of stability between the motion and positional models, even in Figures 14.

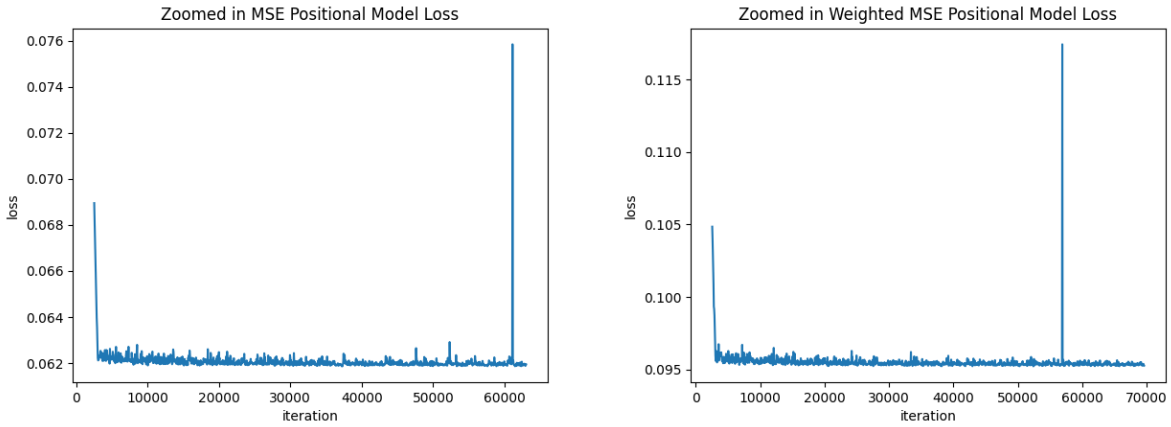


Figure 14: Zoomed in Loss of Positional Models

Figure 14 does also show that the model has either reached its peak performance or is close to it. If the sudden spike is removed, it does appear that the model is becoming more stable and that the noise seen in the loss is becoming smaller. This is similar to what has happened with the motion model.

The positional model MSE results can be seen in Table 6, and it again follows the results seen in the motion model. The main difference is that all of the values are a lot more stable. So even though the model performs worse, it was impacted less by the weighted loss function. Although the weighted loss function does still perform worse. This shows that whatever affected the motion model, likely affected the positional one as well.

Table 6: Mean Square Error Results for Position Models

| Model | Landmarks | Motion |
|---|---|---|
| Positional Weighted MSE | 0.0102660 | 0.0001879 |
| Positional MSE | 0.0098056 | 0.0001791 |

An important thing to notice is that the motion error for the position model is much higher than it was for the motion model. This means that the error frame-to-frame is higher. This is caused by the noise/movement that is seen in the landmarks which are not present in the motion models. This causes the error rate to be higher. If these movements are the model trying to replicate the motion and not random noise, then it is a good demonstration of how the MSE function prefers to produce an average prediction for a lower error, instead of starting to predict the motions. This could mean that although in the loss plots, it appears that the model was near an optimal performance point, a longer training period was necessary.

The analysis for the performance of the different landmarks is the same with as the one stated in the motion model, that the landmarks with a higher error were the ones that would move more such as the hands and fingers. The same goes for the analysis of the input length. In Table 9 in the Append Section 9.3, it can be seen that the performance for sentences between the motion analysis and landmarks analysis is not the same. This is not surprising since the motion difference and landmarks difference are different. The motion MSE is measuring how well the model performed for that frame, while the landmarks MSE is measuring how well it is performing throughout the video. This is because the errors in the positional MSE are compounding. This is also prevalent in the motion Table 10 in the Appendix Section 9.4.

Overall the positional models had a slight difference between them. The positional model without the weighted MSE loss was shown to perform better. The main issue with the positional model was that it would not produce significant motion in the landmarks. The motion that was produced generally appeared to be noise. It was able to create what appeared to be an average position very accurately. This has been due to the loss method being MSE, since an average static figure will produce a lower MSE loss than one that had a lot of motion. That being said the original paper [47] was able to produce motion using an MSE loss. This may mean that the model needs more information than just the positions of the landmarks, since this was provided in their scenarios, or a much longer training period was required.

## 5.3 Summary

The overall summary of the results has shown that neither of the models performed well for this task. Although the positional model would be more useful, due to it producing landmarks that appeared to be closer to a human figure, with the hands sometimes being wrong. The output of a motion vector of zero would not be useful in any scenario. The motion model did have a much better performance frame to frame, as demonstrated by the motion MSE. This is due to the average motion will produce a better result, than the random motions produced by the positional model.

The two models performed similarly for the landmarks MSE. This is due to the random noise motion produced by the landmarks model, which would in most cases average out to zero. This causes it to be equivalent to not producing any motion which the motion model is already doing. As a result, the position models have very similar performance for the models.

In both models, it was shown that the landmarks that represented the extremities performed the worst. This is due to those landmarks having the most amount of movement, which causes them to be the hardest to learn. The expected benefit of using a weighted MSE to try and focus on them more was shown to be detrimental in both models.

A positive about using the motion model is that it appeared to be slightly more stable than the positional model. This is due to the positional model having a large spike in loss

in multiple runs. It was always able to recover back to its previous performance, but it is unclear as to why this happened persistently.

In both models, it was demonstrated that the length of the sentence input appeared to have a small effect on the error. This is due to both models producing no motion. As a result, in both cases, the errors calculated will be from the initial frame of the ground truth of the sentence that is being generated. This means that the errors shown in these measurements for the sentences can be interpreted as how complex the motions are when compared to the initial frame, instead of how accurate the model is.

In the end, the two models performed equivalently for the domains that they were tasked to learn. The positional model, however, is more useful. This is due to it producing the average landmark positions for the sentences. While the motion model only produced a motion vector of zeros, which could be extracted from the positional model. As a result, the positional model provides more information to the user than the motion model would. There is clearly work that can be done to continue improving these models in the future, which will be discussed in Section 7

# 6 Conclusions

The models have been able to demonstrate several things that can be used in the future. The models have shown if the choice is between only motion or position, then the positional model is a better choice to make. Since the training only on motion model will only give you vectors of zeros, which would be useless without the initial frame of the video. It also appears that the dataset used may have been too complex. A simpler dataset with fewer words would have allowed the model to see more instances of each translation from the word to gloss in the sentences. Lastly, the weighted MSE loss was not helpful for this model. The assumption of focusing on certain landmarks did not appear to be helpful. These observations and how they can be implemented will be further discussed in this section.

The first observation was that the positional model was a better model to choose if the choice is only between motion and positional models. Even though the performance of the motion model was better when looking at its performance frame to frame, and was very similar when comparing the performance across the entire video. This is all out weighted the motion model does not provide any actual information that could be used. The positional model will at least provide the user with the average position of landmarks for the given sentence.

The next observation was that a simpler and more repetitive dataset would be better to use than a diverse one. This means that the RWTH-PHOENIX-Weather dataset [19] is likely a better dataset to use in future papers, such as this one. Since the words being used are more likely to be repeated and similar sentences to be used. The large diversity of the words being used in this dataset, likely made it harder for the model to learn how

to output the words that it had seen multiple times. A way to get around this would be adding augmentation. Although as stated previously, the augmentation being used needs to be carefully selected. Since many of the commonly used augmentations for images or video generation, would not be applicable in this scenario.

The last observation is about the loss functions being used. As mentioned the main comparison was a regular MSE loss function to a weighted MSE function. The reasoning for using the weighted MSE function was that the model would be forced to focus on landmarks that have a lot of motion. These landmarks are more likely to produce a larger error measurement. This was shown not to work, since in both cases the weighted MSE models performed worse. It is possible that the weights were set incorrectly, thus making the model focus too much on these landmarks and not giving enough attention to the rest of the landmarks. Another important note is that the positional model was creating a video of essentially static landmarks. This was also likely due to MSE loss function. Since the model learned that static landmarks of the average positions will produce lower errors than trying to predict the motion of the landmarks. Interestingly the weighted MSE did not seem to have a strong influence on the positional model.

Overall, the thesis has provided several important notes for future projects in video generation of sign language using text inputs. Although usable results were not produced in this thesis, it is clear that it would be possible to use the MotionDiffuse or another DDPM model for this type of problem. It is better to focus on the positional features of the data than the motion parts. This will allow the model to at least provide an average position, from which it may continue learning the motion. It is also important for the dataset to contain strings of words repetitively. This will make it easier for the model to learn how to transition between words, as well as learn the words themselves. Lastly, although the MSE loss function worked in the original paper [47], other loss functions should be tested to avoid static landmarks. In conclusion, a DDPM model is a very likely future candidate for generating well-produced videos of sign interpreters.

# 7 Future Work

There are several things that can be done in the future to help provide better results. The output vectors that the model is creating could be more detailed. An example of this would be combining the motion vectors calculated from the dataset and position landmarks provided by the dataset. It would also be helpful to try allowing for longer videos or testing another dataset. The filtering applied to the LSFB dataset [8] severely limited how many instances it was possible to use. Further experimentation of the parameters will also likely prove to be beneficial. All of these possible improvements will be discussed in this section.

The first improvement mentioned is the output vectors being more detailed. Although it is unlikely that adding more landmarks would provide better results, it is likely that a combination of the motion and positional values would provide better results. The

additional information may allow the model to create better associations between the landmarks and their movements. That being said since both models tended to go to the average, it is also possible that the model will just become better at producing a static figure of the average landmarks with less noise being present.

The next improvement is providing more data to the model. The current dataset was filtered significantly in order to limit how many frames were being produced. If this limitation is lifted, then the model would have a lot more data that it would be able to access in the dataset. This would give it more possibilities to encounter the same words again. The other solution to this issue would be using another dataset. As mentioned the RWTH-PHOENIX-Weather dataset [19] is a commonly used dataset. Since it is focused on the weather, the range of expressions used is likely to be much lower. As a result, it is much more likely to encounter the same words and phrases, helping the model to learn better. Lastly, the current data could be augmented. This would artificially increase the dataset. Overall, the model should encounter some words and phrases more often. This increase by either artificially increasing the dataset or finding a dataset where this happens more commonly would likely increase the performance significantly.

The last improvement would be continuing to experiment with the parameters of the model, such as the loss function. It is possible that in the current thesis, too much focus was placed on certain landmarks, which caused the results for the weighted MSE to be worse than for the regular MSE. It should be tested if lowering these weights would produce better results. It should also be tested if using other loss functions, in general, could produce better results as well. This may also help overcome the static landmarks experienced with the positional model. It would also be beneficial to test a much longer training period for the models. DDPM models generally take a while to train, and the thesis ended the training due to it appearing that the model has finished learning. It is possible that this was not the case and further training was necessary.

As mentioned in the introduction, the eventual landmarks produced by the model could be used to generate a video of a sign language interpreter. There are various ways of doing this, such as using style transfer [17] [29] or inputting it in some other type of model to try and create the videos. There are already existing programs that take landmarks and will move a CGI human model according to them. This can be seen CGI videos that use motion tracking. The video of this CGI human moving can then be fed through a style transfer model to try and transfer it to a real video would likely produce good results.

This thesis has clearly provided a good step forward in translating text into a video of sign language. Although a lot of work has been done, there are still other things that can be tested to help improve the performances of the models. These improvements are all likely to help continue improving the performance of the models tested. As mentioned before, with the popularity of the new DDPM models, it is very likely that one of them will be the next large improvement in this area.

# 8 Potential Negative Impacts

As with any artificial intelligence project, it is important to take into account what negative impacts this could have on society. If this model was developed completely to generate realistic-looking videos that could be used by the public, the main concern would be the smaller demand for sign language interpreters. Since it is likely that companies would be more likely to replace them with videos. That being said, this may be outweighed by the increase in accessibility for the people who would be using these videos.

A much larger issue comes from the indirect problem of using this model. If the model is able to create realistic-looking videos from text input, these could be used to generate deep fakes or misinformation. It is important to note that several additional steps would have to be done in order for this to happen, such as either having style transfer applied to create a realistic video or further training the model to use the inputs for video generation. In either case, this could be a problematic usage of the model that could become widely available.

The solutions to this all have their own flaws and it is difficult to know what will be the correct response. A model trained to detect AI-generated videos will only work until another model comes along that creates a video that can fool it, and this may end up working as a GAN model, each learning to become better. If a model is released for public use, it may be possible to code it to prevent it from making videos of public figures. This solution only would work if the public does not have access to the model itself, but only has access to it by giving it inputs. As soon as someone gets the actual model themselves, they would be able to produce their own videos without this limitation.

This is an important topic that the AI community needs to discuss how to handle properly. Since the damage that could be done is immense, it is important to take this into account. That being said, research in this field should continue to help better understand how videos can be generated and to be able to find new positive applications of the models to help improve society.

# References


[1] Amy LC Follow High School Social Studies Teacher at Program for Deaf and Hard of Hearing. Glossing in asl. what is it? eight examples.

[2] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. *CoRR*, abs/2111.14818, 2021.

[3] Kshitij Bantupalli and Ying Xie. American sign language recognition using deep learning and computer vision. In *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, December 2018.

[4] German Barquero, Sergio Escalera, and Cristina Palmero. Belfusion: Latent diffusion for behavior-driven human motion prediction, 2022.

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020.

[6] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, Jingyi Yu, and Gang Yu. Executing your commands via motion diffusion in latent space, 2022.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[8] Jérôme Fink, Benoît Frénay, Laurence Meurant, and Anthony Cleve. Lsfb-cont and lsfb-isol: Two new datasets for vision-based sign language recognition. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021.

[9] Rui Gao, Xingsong Hou, Jie Qin, Jiaxin Chen, Li Liu, Fan Zhu, Zhao Zhang, and Ling Shao. Zero-vae-gan: Generating unseen features for generalized and transductive zero-shot learning. *IEEE Transactions on Image Processing*, 29:3665–3680, 2020.

[10] Anthony Gillioz, Jacky Casas, Elena Mugellini, and Omar Abou Khaled. Overview of the transformer-based models for NLP tasks. In *Proceedings of the 2020 Federated Conference on Computer Science and Information Systems*. IEEE, September 2020.

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[12] Shir Gur, Sagie Benaim, and Lior Wolf. Hierarchical patch vae-gan: Generating diverse videos from a single sample. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16761–16772. Curran Associates, Inc., 2020.

[13] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos, 2022.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020.

[15] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models, 2022.

[16] Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling, 2022.

[17] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *IEEE Transactions on Visualization and Computer Graphics*, 26(11):3365–3385, 2020.

[18] Christopher Kissel, Christopher Kümmel, Dennis Ritter, and Kristian Hildebrand. Pose-guided sign language video GAN with dynamic lambda. *CoRR*, abs/2105.02742, 2021.

[19] Oscar Koller, Jens Forster, and Hermann Ney. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141:108–125, December 2015.

[20] Jinyu Li, Yu Wu, Yashesh Gaur, Chengyi Wang, Rui Zhao, and Shujie Liu. On the comparison of popular end-to-end models for large scale speech recognition. *arXiv preprint arXiv:2005.14327*, 2020.

[21] Yitong Li, Martin Min, Dinghan Shen, David Carlson, and Lawrence Carin. Video generation from text. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[22] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11461–11471, June 2022.

[23] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics:*

*System Demonstrations*, pages 55–60, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[24] Lu Mi, Macheng Shen, and Jingzhao Zhang. A probe towards understanding GAN and VAE models. *CoRR*, abs/1812.05676, 2018.

[25] Gustav Müller-Franzes, Jan Moritz Niehues, Firas Khader, Soroosh Tayebi Arasteh, Christoph Haarburger, Christiane Kuhl, Tianci Wang, Tianyu Han, Sven Nebelung, Jakob Nikolas Kather, and Daniel Truhn. Diffusion probabilistic models beat gans on medical images, 2022.

[26] B. Natarajan, R. Elakkiya, and Moturi Leela Prasad. Sentence2signgesture: a hybrid neural machine translation network for sign language video generation. *Journal of Ambient Intelligence and Humanized Computing*, January 2022.

[27] Ilias Papastratis, Kosmas Dimitropoulos, Dimitrios Konstantinidis, and Petros Daras. Continuous sign language recognition through cross-modal alignment of video and text embeddings in a joint-latent space. *IEEE Access*, 8:91170–91180, 2020.

[28] Saeed Saadatnejad, Ali Rasekh, Mohammadreza Mofayezi, Yasamin Medghalchi, Sara Rajabzadeh, Taylor Mordan, and Alexandre Alahi. A generic diffusion-based approach for 3d human pose prediction in the wild, 2022.

[29] Hiroshi Sasaki, Chris G. Willcocks, and Toby P. Breckon. UNIT-DDPM: unpaired image translation with denoising diffusion probabilistic models. *CoRR*, abs/2104.05358, 2021.

[30] Ben Saunders, Necati Camgoz, and Richard Bowden. Everybody sign now: Translating spoken language to photo realistic sign language video. 11 2020.

[31] Ben Saunders, Necati Cihan Camgöz, and Richard Bowden. Adversarial training for multi-channel sign language production. *CoRR*, abs/2008.12405, 2020.

[32] Ben Saunders, Necati Cihan Camgöz, and Richard Bowden. Progressive transformers for end-to-end sign language production. *CoRR*, abs/2004.14874, 2020.

[33] Zhuoran Shen, Mingyuan Zhang, Shuai Yi, Junjie Yan, and Haiyu Zhao. Factorized attention: Self-attention with linear complexities. *CoRR*, abs/1812.01243, 2018.

[34] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

[35] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

[36] Stephanie Stoll, Necati Camgoz, Simon Hadfield, and Richard Bowden. Sign language production using neural machine translation and generative adversarial networks. 08 2018.

[37] Stephanie Stoll, Necati Cihan Camgoz, Simon Hadfield, and Richard Bowden. Text2sign: Towards sign language production using neural machine translation and generative adversarial networks. *International Journal of Computer Vision*, 128(4):891–908, January 2020.

[38] Hamid Vaezi Joze and Oscar Koller. Ms-asl: A large-scale data set and benchmark for understanding american sign language. In *The British Machine Vision Conference (BMVC)*, September 2019.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[40] Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. Mcvd: Masked conditional video diffusion for prediction, generation, and interpolation, 2022.

[41] Tsun-Hsuan Wang, Yen-Chi Cheng, Chieh Hubert Lin, Hwann-Tzong Chen, and Min Sun. Point-to-point video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[42] Dong Wei, Huaijiang Sun, Bin Li, Jianfeng Lu, Weiqing Li, Xiaoning Sun, and Shengxiang Hu. Human joint kinematics diffusion-refinement for stochastic motion prediction, 2022.

[43] Lilian Weng. What are diffusion models?, Jul 2021.

[44] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using VQ-VAE and transformers. *CoRR*, abs/2104.10157, 2021.

[45] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation, 2022.

[46] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[47] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model, 2022.

# 9 Appendix

## 9.1 Total Motion Results

Table 7: Landmark Results for Motion Models

| | Motion MSE Model | | Weighted Motion MSE Model | |
|---|---|---|---|---|
| Landmark | MSE Motion | Landmarks | MSE Motion | Landmarks |
| total mse | 9.87E-05 | 0.009673394502311 | 0.391549921001031 | 940.225780532215 |
| RIGHT WRIST | 8.86E-05 | 0.01023022076022 | 0.272814269648698 | 767.379735401712 |
| RIGHT THUMB TIP | 0.000174123749372 | 0.018047326235464 | 0.247765675711463 | 693.058021933331 |
| RIGHT INDEX FINGER TIP | 0.000230320500651 | 0.025045007946841 | 0.632462452880065 | 1086.29517202941 |
| RIGHT MIDDLE FINGER TIP | 0.000252066994921 | 0.022107253117982 | 0.593030674636597 | 1087.84532713707 |
| RIGHT RING FINGER TIP | 0.000209527817701 | 0.021148403753053 | 0.598378022173913 | 1108.81948265348 |
| RIGHT PINKY TIP | 0.00021049686352 | 0.019894791505267 | 0.573207193627808 | 1101.84963505561 |
| LEFT WRIST | 5.37E-05 | 0.006855442356554 | 0.476512977866555 | 1286.61901189408 |
| LEFT THUMB TIP | 0.000109942902666 | 0.01295341506463 | 0.464615674504175 | 1133.62071242871 |
| LEFT INDEX FINGER TIP | 0.000145435286386 | 0.017505357223906 | 0.48708302872947 | 1131.01685190577 |
| LEFT MIDDLE FINGER TIP | 0.000132091540209 | 0.016204409090645 | 0.494187828989157 | 1199.38777529564 |
| LEFT RING FINGER TIP | 0.000120186259047 | 0.014921329876356 | 0.513376835428 | 1235.90624636749 |
| LEFT PINKY TIP | 0.000112263039587 | 0.013449170249033 | 0.517089172273867 | 1273.93232395023 |
| NOSE | 2.08E-05 | 0.000426990551225 | 0.185281341637733 | 517.314118160442 |
| LEFT EAR | 2.02E-05 | 0.00031371789607 | 0.199107452296146 | 558.347996586131 |
| RIGHT EAR | 2.28E-05 | 0.000299705716302 | 0.155096252837659 | 431.955258953208 |
| MOUTH LEFT | 1.85E-05 | 0.000362302574596 | 0.208952392432849 | 581.624187815202 |
| MOUTH RIGHT | 1.97E-05 | 0.000374943171363 | 0.185776186148355 | 522.532951272828 |
| LEFT SHOULDER | 1.56E-05 | 0.000243216762129 | 0.317410194588999 | 890.559336208596 |
| RIGHT SHOULDER | 2.56E-05 | 0.000254555142326 | 0.200645851890655 | 562.232607814236 |
| LEFT ELBOW | 3.77E-05 | 0.001179563385914 | 0.530502800418344 | 1512.0454262256 |
| RIGHT ELBOW | 5.27E-05 | 0.001324162168646 | 0.369252062301149 | 1062.39921208772 |

## 9.2 Total Landmark Results

Table 8: Landmark Results for Positional Models

| | Positional MSE Model | | Weighted MSE Positional Model | |
|---|---|---|---|---|
| Landmark | MSE Motion | Landmarks | MSE Motion | Landmarks |
| total mse | 0.000179084237919 | 0.009805620988127 | 0.000187960144504 | 0.010266001215951 |
| RIGHT WRIST | 8.55E-05 | 0.009981920200438 | 8.53E-05 | 0.009991065509628 |
| RIGHT THUMB TIP | 0.000168230024925 | 0.017608308913397 | 0.000167799807486 | 0.017617576784046 |
| RIGHT INDEX FINGER TIP | 0.000471909357029 | 0.02503002982785 | 0.000447929881472 | 0.024678359814923 |
| RIGHT MIDDLE FINGER TIP | 0.000423358762845 | 0.023401382472445 | 0.000395834241933 | 0.023485284100761 |
| RIGHT RING FINGER TIP | 0.000371986446513 | 0.021746436418335 | 0.000357015363259 | 0.021864727784052 |
| RIGHT PINKY TIP | 0.000365761730151 | 0.020602609286872 | 0.000361900378559 | 0.020571321416785 |
| LEFT WRIST | 0.000119982453042 | 0.006956373527659 | 0.000152972965956 | 0.0076677384355 |
| LEFT THUMB TIP | 0.000235497566455 | 0.01327970734189 | 0.000279657582348 | 0.014665364630521 |
| LEFT INDEX FINGER TIP | 0.000333975175834 | 0.017440613209679 | 0.000373524403447 | 0.019667801215067 |
| LEFT MIDDLE FINGER TIP | 0.000285657677106 | 0.016113625151465 | 0.000300614667281 | 0.01831906128764 |
| LEFT RING FINGER TIP | 0.000272537112138 | 0.015153892312639 | 0.000308491130348 | 0.016754094439714 |
| LEFT PINKY TIP | 0.000246474572988 | 0.013728831098715 | 0.000287690084051 | 0.015385542876964 |
| NOSE | 2.92E-05 | 0.000440718616397 | 3.66E-05 | 0.000441302299263 |
| LEFT EAR | 3.61E-05 | 0.000324539346899 | 3.89E-05 | 0.000331831432962 |
| RIGHT EAR | 3.71E-05 | 0.000313144709 | 4.36E-05 | 0.000318639080985 |
| MOUTH LEFT | 2.89E-05 | 0.000368842571736 | 3.37E-05 | 0.000371125380678 |
| MOUTH RIGHT | 2.90E-05 | 0.0003804055304 | 3.34E-05 | 0.000380115083403 |
| LEFT SHOULDER | 3.03E-05 | 0.000254714106518 | 3.97E-05 | 0.000259241702635 |
| RIGHT SHOULDER | 3.91E-05 | 0.000269061423357 | 4.50E-05 | 0.000290855150933 |
| LEFT ELBOW | 6.50E-05 | 0.001167249395713 | 7.16E-05 | 0.001186400408929 |
| RIGHT ELBOW | 8.52E-05 | 0.001355635289275 | 8.59E-05 | 0.001338576699591 |

## 9.3 Postional Model Sentence Results

Table 9: Sentence Analysis for Positional Models

| Weighted MSE Positional Model | | | | | | | |
|---|---|---|---|---|---|---|---|
| Motion | | | | Landmarks | | | |
| Lowest | | Highest | | Lowest | | Highest | |
| input | mse | input | mse | input | mse | input | mse |
| oui | 6.27E-06 | à toi | 7.79E-04 | je ne sais... | 1.31E-04 | non mais... | 2.46E-02 |
| à moi | 7.04E-06 | cest différent... | 8.14E-04 | non | 2.49E-04 | et aussi... | 2.57E-02 |
| je ne sais pas | 7.27E-06 | tous les jours | 8.39E-04 | oui léquivalent... | 3.70E-04 | on a lhabitude | 2.72E-02 |
| non je ne sais... | 1.31E-05 | ah oui son nom | 8.86E-04 | non je ne sais... | 5.11E-04 | aussi parfois | 3.54E-02 |
| cest vrai il nest... | 2.18E-05 | bruxelles mais... | 1.15E-03 | oui cest vrai | 5.17E-04 | oui oui | 3.56E-02 |
| non | 2.39E-05 | ah ça | 1.28E-03 | cest perdu | 5.19E-04 | jai changé... | 3.85E-02 |
| oui oui oui voilà... | 2.48E-05 | voilà | 1.28E-03 | cest vrai | 5.34E-04 | ah oui jen ai... | 5.15E-02 |

| Weighted MSE Positional Model | | | | | | | |
|---|---|---|---|---|---|---|---|
| Motion | | | | Landmarks | | | |
| Lowest | | Highest | | Lowest | | Highest | |
| input | mse | input | mse | input | mse | input | mse |
| non | 9.63E-06 | pour moi | 6.29E-04 | non | 2.63E-04 | non mais ça... | 2.60E-02 |
| pour les sourds... | 1.86E-05 | moi oui | 6.32E-04 | à moi | 2.81E-04 | avec ma... | 2.65E-02 |
| oui il doit être... | 1.97E-05 | tu en as de... | 7.27E-04 | je ne sais pas | 3.24E-04 | on a lhabitude | 3.38E-02 |
| comme vous | 2.11E-05 | aussi parfois | 7.32E-04 | cest perdu | 3.35E-04 | aussi parfois | 3.63E-02 |
| oui léquivalent du... | 2.25E-05 | oh cétait facile | 8.54E-04 | non je ne sais... | 3.69E-04 | oui oui | 3.89E-02 |
| oui il y en a trois | 2.25E-05 | cest juste | 8.72E-04 | oui | 4.53E-04 | jai changé... | 3.93E-02 |
| oui oui oui voilà oui | 2.37E-05 | bruxelles mais... | 1.16E-03 | oui cest vrai | 4.85E-04 | ah oui jen ai... | 4.84E-02 |

## 9.4 Motion Model Sentence Results

Table 10: Sentence Analysis for Landmark Models

| Motion Model Weighted Loss | | | | | | | |
|---|---|---|---|---|---|---|---|
| Motion | | | | Landmarks | | | |
| Lowest | | Highest | | Lowest | | Highest | |
| input | mse | input | mse | input | mse | input | mse |
| vasy | 1.66E-01 | tous les jours | 8.60E-01 | vasy | 2.59E+00 | attention le suivi... | 5.16E+03 |
| je nai jamais vu | 1.90E-01 | moi aussi | 8.70E-01 | cest perdu | 5.83E+00 | ah tu as appris... | 5.24E+03 |
| je pense que | 1.92E-01 | ah ça | 9.76E-01 | super | 8.52E+00 | daccord pas... | 5.40E+03 |
| ah oui | 2.30E-01 | cest différent pour moi | 9.99E-01 | cest juste | 1.45E+01 | oui on a lhabitude | 5.61E+03 |
| quoi lécole de lirsa quoi | 2.32E-01 | cest vrai | 1.19E+00 | à mon tour | 1.66E+01 | toi tu étais à... | 6.01E+03 |
| attention le suivi nest... | 2.39E-01 | cest juste | 1.26E+00 | je pense que | 1.82E+01 | voilà la différence | 7.01E+03 |
| on en parle après | 2.46E-01 | la pollution | 1.33E+00 | elle y pense déjà | 2.28E+01 | non mais ça... | 1.39E+04 |

| Motion Model | | | | | | | |
|---|---|---|---|---|---|---|---|
| Motion | | | | Landmarks | | | |
| Lowest | | Highest | | Lowest | | Highest | |
| input | mse | input | mse | input | mse | input | mse |
| oui | 9.43E-07 | a toi | 2.71E-04 | je ne sais pas | 4.63E-05 | et aussi travailler... | 2.52E-02 |
| à moi | 2.80E-06 | comment faire | 2.95E-04 | non | 1.69E-04 | a toi | 2.61E-02 |
| je ne sais pas | 3.46E-06 | ah ça | 3.10E-04 | non je ne sais pas | 1.86E-04 | on a lhabitude | 3.10E-02 |
| non | 5.59E-06 | après on nous dit ce... | 3.20E-04 | à moi | 3.51E-04 | oui oui | 3.83E-02 |
| non je ne sais pas | 5.99E-06 | cest juste | 4.48E-04 | oui | 4.38E-04 | aussi parfois | 3.88E-02 |
| voilà | 1.25E-05 | moi oui | 5.51E-04 | oui cest vrai | 5.25E-04 | jai changé deux... | 4.45E-02 |
| cest vrai il nest pas facile | 1.64E-05 | bruxelles mais aussi namur | 1.17E-03 | oui léquivalent du français signé | 5.25E-04 | ah oui jen ai un... | 5.24E-02 |