Motion Diffusion Model to Denoising Diffusion GAN: Efficient Motion Sampling

Ronald Campos
University of Central Florida
4000 Central Florida Blvd, Orlando, FL 32816
roncamposj@knights.ucf.edu

Muhammad Asad Haider
University of Central Florida
4000 Central Florida Blvd, Orlando, FL 32816
haider24@knights.ucf.edu

Suneet Tipirneni
University of Central Florida
4000 Central Florida Blvd, Orlando, FL 32816
suneet.tipirneni@knights.ucf.edu

Stefan Werleman
University of Central Florida
4000 Central Florida Blvd, Orlando, FL 32816
stefanwerleman@knights.ucf.edu

Abstract

All existing motion diffusion models use the standard diffusion process which yields high quality samples. However, the standard process for these models can be inefficient. These are one of the challenges with the learning trilemma and this works concerns embedding an existing motion diffusion model into denoising diffusion GANs to create a hybrid architecture of the motion diffusion model. This new hybrid model will satisfy the learning trilemma, thus improving the sampling speed when training and generating a motion sample. https://github.com/CAP6412-Group-4/denoising-diffusion-gan

1. Introduction

Deep generative models had many breakthroughs in past years. Many applications have been built such as: image synthesis, inpainting, image classification, segmentation, point clouds, and audio. One of the latest developments with these models is the ability to generate human motions based on text inputs. However, there are three key requirements that generative models cannot satisfy: high-quality sampling, mode coverage and diversity, and fast sampling.

1.1. Human Motion Diffusion Model

The human motion diffusion model (MDM) is a generative model that generates human motions. Given a text prompt that describes the motion the applicatin would output a video that shows a skeleton figure perfoming the action that was described in the text prompt.

The MDM can generate high quality motion samples and achieve good mode coverage. However, the generative model for MDM requires a great amount of timesteps in the reverse process which means that the current MDM does not satisfy the fast sampling in the learning trilemma (Figure **??**).

1.2. Improving Sampling

1.3. Integrating Motion Diffusion Model Into DDGAN

2. Related Work

2.1. Human Motion Diffusion Model

2.2. Denoising Diffusion GANs

3. Method

3.1. Motion Diffusion Model Integration

As implied by the title and previous text, our model aims to apply the sampling gains provided by [**?**]. However in doing this we needed to make fundamental changes to how the MDM is constructed. More specifically, The model needs to ingest the latent z variable as decribed in [**?**]. Secondly, the DDGAN model needs to adjust it's dimensionality to take in frames of joint positions rather than taking a single image frame dimensions as input.

3.1.1 MDM Modifications

The MDM model as-provided already provides a fantastic framework for a generator model. Most of the model can remain intact however additions were made to make the model more versatile. The primary change occurs with the introduction of the latent z variable. This variable is the mapping variable used for conditional score networks and allows the MDM model to sample from a non-normal distribution [**?**]. The latent z variable is added to the

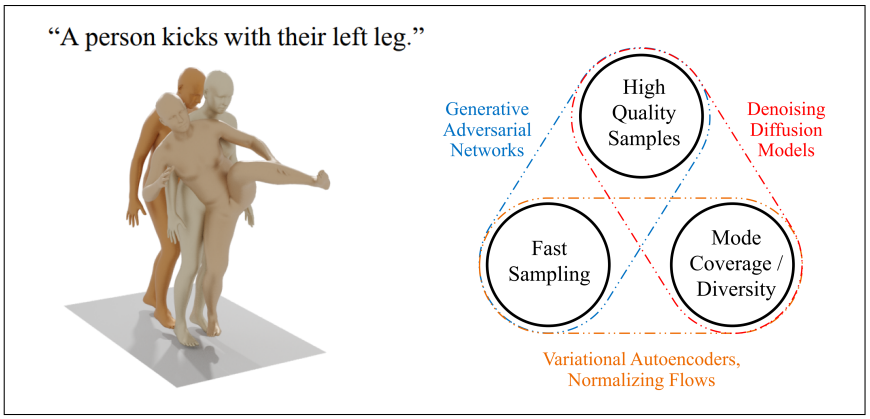

Figure 1. Motion synthesise and the Generative learning trilemma [?] [?]

other inputs described in [?]. For our our purposes we use different z mapping layers that discard image-specific normalization and change output channels to match the desired dimenions of the humanml dataset. Our resulting MDM architecture is shown if figure **??**.

3.1.2 DDGAN Modifications

As for the denoising-diffusion-GAN (ddgan). We fixed the image dimension of the model to 120. This is done to align with the dimensionality used in [?]. Instead of using the generator provided by [?] we replace the generator with our modified MDM architecture. As a result we are effectively using the discrminator to discriminate against samples our MDM model generates.

3.2. Adapting The Loss

As described in **??** we are using the discriminator to validate the values given from our generator. The discrinator offers us adversarial loss, however, this loss along is not sufficient for our generator. Our generate being an MDM requires additional losses to produce high-quaility outputs. For this we borrow the geometric losses described in [?]. Even though adversarial loss is not sufficient to properly to the model alone, it is still needed as a loss to propogate. As a result, we simply add adversarial loss to the geometric losses which results in equation **??**.

$$\mathcal{L}_{all} = \lambda_{adv}\mathcal{L}_{adv} + \lambda_{pos}\mathcal{L}_{pos} + \lambda_{vel}\mathcal{L}_{vel} + \lambda_{foot}\mathcal{L}_{foot} \quad (1)$$

4. Experiments

Our implementation of MDM-2-Diffgan (M2D) is applied for the task of text-to-motion generation. We trained our models with T = 4 noising steps, using a cosine noise schedule. Experiments were conducted using the Newton cluster which employs two NVIDIA V100 GPUs per node. During the training process of our models, we conducted experiments with varying numbers of epochs, ranging from 200 to 1200, and determined that the most optimal results were achieved with fewer epochs, specifically at the lower end of the range. Training on this lower range took approximately about a day.

The HumanML3D dataset, which was also used by [?], was employed in our text-to-motion generation task. HumanML3D [?] is a combination the HumanAct12 [?] and Amass [?] datasets, which covers a broad range of activities, such as 'jumping', dancing', and certain other acrobatics. The dataset contains 14,616 motion clips and 44,970 text descriptions.

Our models were trained with a batch size of 128. Our generator used a learning rate of 0.0015, while the learning rate of the discriminator was slightly larger at 0.0001. We employed the same parameters as [?] did with the transformer encoder and used that as our generator: 8 layers, an embedding dimension of 512, a GELU activation function, and dropout with a rate of 0.1. The discriminator uses an embedding dimension of 128, to go along with its 6 downsampling blocks. Its final output is 256 channels. Both the generator and discriminator use a softplus loss function, which is a smooth aproximation of the ReLU. We conducted experiments with varying weights λ for the geometric losses, but observed that our results were unsatisfactory when $\lambda > 0$, which aligns with the findings reported in [?]. Geometric losses are already represented in the HumanML3D ensemble, thus it is justifiable to omit them during training.

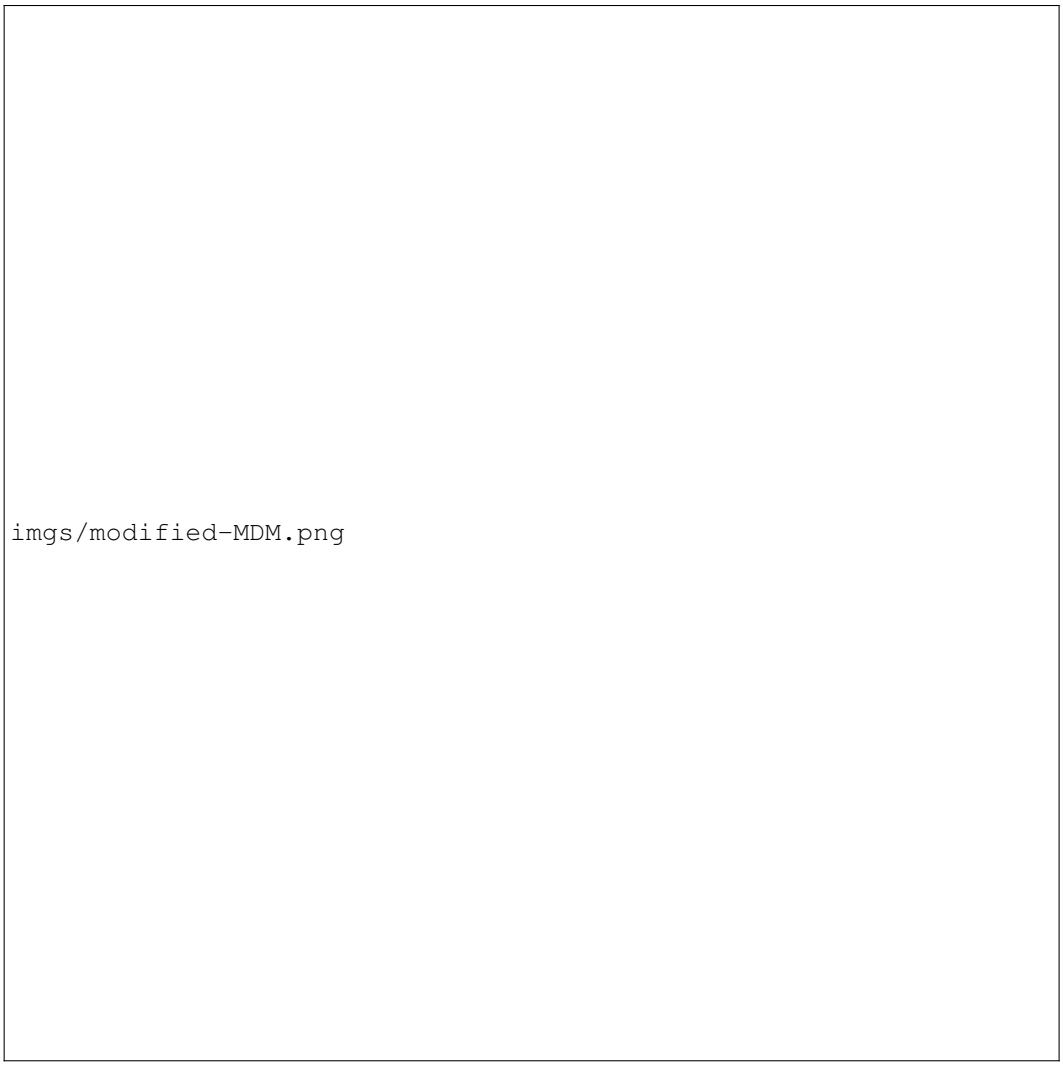

Figure 2. Our modified MDM architecture based on [**?**], we introduce the latent z variable by adding it with the timestep condition and the text prompt condition c

5. Results

5.1. Qualitative Results

Our qualitative findings illustrate that our model is capable of most generations that MDM can perform, but with a few discrepancies we should note. In the final seconds of our generations, our results will generally exhibit a floor sliding effect, where after following the text prompt, the plane beneath the figure will shift. This is not the same as the foot contact sliding effect, since by this point the prompt has been completed and the generation is just standing still. This effect is not present in the MDM generations, and we believe that this is due to the fact that the MDM model is trained on a dataset that does not contain this effect. We believe this is a result of some model instability, but at this time are unable to pinpoint the exact cause.

Taking this into consideration, we maintain our conviction that the generations excel in capturing what the text-prompts indicate. We believe that the distribution of HumanML3D is effectively captured by MDM-2-Diffgan, where we observed a favorable range of variability in generations. In the future, we would like to perform a user study in order to obtain less biased interpretaions of our results.

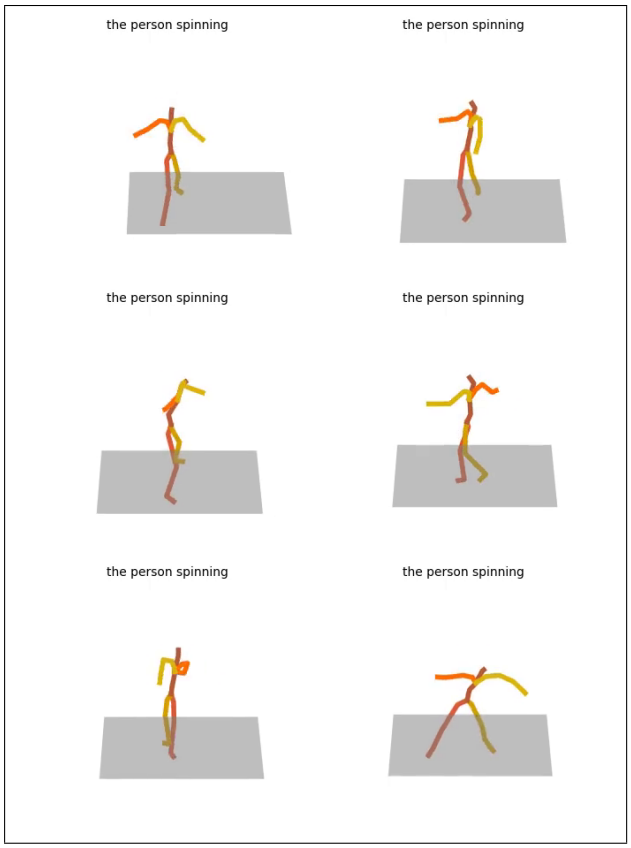

Figure 3. The shows a person in the act of spinning. (Subject to change)

Method	R Precision (top 3)↑	FID ↓	Multimodal Dist ↓
Real	$0.797_{\pm.002}$	$0.002_{\pm.000}$	$2.974_{\pm.008}$
JL2P	$0.486_{\pm.002}$	$11.02_{\pm.046}$	$5.296_{\pm.008}$
T2M	$0.740_{\pm.003}$	$1.067_{\pm.002}$	$3.340_{\pm.008}$
MDM	$0.611_{\pm.007}$	$0.544_{\pm.044}$	$5.566_{\pm.027}$
M2D	$0.698_{\pm.007}$	$2.44_{\pm.429}$	$6.353_{\pm.076}$

Table 1. MDM-2-Diffgan accomplishes a higher R-Precision score than MDM, but a substantially lower FID score.

Method	Diversity →	Multimodality ↑
Real	$9.503_{\pm.065}$	-
JL2P	$7.676_{\pm.058}$	-
T2M	$9.188_{\pm.002}$	$2.090_{\pm.083}$
MDM	$9.559_{\pm.086}$	$2.799_{\pm.072}$
M2D	$9.416_{\pm.057}$	$2.671_{\pm.025}$

Table 2. We observe a similarity of around 0.1 for both metrics between MDM and MDM-2-Diffgan.

Method	1s & 1r	3s & 1r	10s & 3r
	Seconds		
MDM	16.58	18.28	105.14
M2D	0.22	0.31	0.64

Table 3. Our samples generated around 100x faster than those from MDM.

5.2. Quantitative Results

As a quick reminder, let's recall the metrics we use to evaluate our model. We use the R-Precision score, which measures the similarity between the generated motion and the text prompt. FID and Multimodal Distance scores measures the similarity between the generated motion and the real motion. Lastly, Diversity measures the variability between the generated motions.

We compare our results with the results of [**?**] and other motion generation models, which are shown in Table **??** and **??**. We can observe that MDM-2-Diffgan performs fairly well when compared to MDM. While we don't set the state-of-the-art the state of the art in any of the metrics, we are able to achieve a higher R-Precision score than MDM. Also we have a comparable diversity and Multimodality scores, which is a good sign that our model is able to capture the distribution of the HumanML3D dataset. We observe a poor FID score when compared to MDM.

We also perform a speed test, where we measure the time it takes MDM-2-Diffgan to generate motions. We perform three experiments where we generate motions with: 1 sample s and 1 repetition r, 3 samples s and 1 repetition r, and 10 samples s and 3 r. From the results in Table **??**, we can observe that it takes MDM considerably longer to generate motions than Motion-2-Diffgan. We can see that MDM takes 16.58 seconds to generate 1 sample and 1 repetition, while M2D takes 0.22 seconds. What is noteworthy though, is how both models scale in as we increase the number of samples and repetitions. MDM takes 105.14 seconds to generate 10 samples and 3 repetitions, whereas M2D only takes 0.64 seconds. This is a significant improvement in time, and shows that MDM-2-Diffgan is able to scale much more effectively than MDM.

6. Additional Applications

Once our hybrid motion diffusion model was complete and generating results, we want to see how we can leverage the motion samples for other applications.

6.1. Using Motion Samples for Person Image Synthesis (PIDM)

Figure 4. Image sample of the pose-guided image synthesis.

We wanted to use the produced motion samples from our hybrid as target poses or position to generate photorealistic images of humans [**?**]. There are several pose-guided person image generation models but this model proposed by Ankan Bhunia et al. (2023) utilizes a denoising diffusion model to generate the image samples (Figure **??**).

The process goes as follows: given a target pose in the form of a skeleton and a reference image of a person, in each diffusion step the model generates a sharper version of the final image until it reaches T total diffusion steps (Figure **??**). The target pose is a color-coded where each color represents a joint or body of the person in the final sampled image. For instance, the green joints in the skeleton represents the position of the head, the yellow shoulder represents the left shoulder position, and so on.

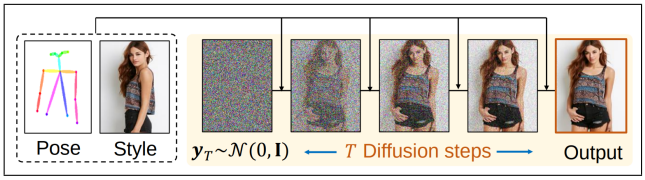

Figure 5. General workflow for the Person Image Synthesis Denoising Diffusion Model.

In Figure **??**, our motion diffusion model produces skeletons that perform certain actions. However, our skeleton images are color-coded differently and are missing the head position of the person generated. First step was to determine a mapping from the motion diffusion skeleton to the correct color-codes for PIDM. Next, is to take the target pose image sample and create a numpy array representation of it. The PIDM diffusion model expected an input shape of 256x256x20 where the first 3 channels represent the pose skeleton while the remaining 17 skeletons are gaussian keypoint maps. We were only able to reproduce the first 3 channels from our MDM skeleton pose and created the remaining 17 channels with values of zeros.

Figure 6. The target pose generated by our motion diffusion model was suppose to represent a person who is facing their back towards us.

After, getting our custom pose image to the appropraite array representation for the diffusion model in PIDM, we were able to finally test it and generate a sample pose-guided image. Unfortunately, the sampled image did not produce the results we expected as the final person image appeared to be disfigured (Figure **??**). This is probably due to the absence of the 17 guassian keypoint channels in our target pose skeleton and is also due to the missing head position in the target pose.

7. Conclusion and Future Works