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

Human motion modeling is important for many modern graphics applications. It is a challenging task that requires skill and time. Recently, several motion generation models have been proposed to automate this process. However, these models are not able to satisfy the generative trilemma: high-quality sampling, mode coverage and diversity, and fast sampling. In this paper, we propose a novel generative model that utilizes a hybrid diffusion process to generate human motions. We are able to improve the sampling speed by upto 100x over the state of the art approach while maintaining the quality of the generated motions. 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. This issue is more commonly referred to as the learning trilemma and it is an issue that has plagued many generative models over the past years. With the introduction of [3] we see promising solution to this for the motion predition domain.

1.1. Human Motion Diffusion Model

The human motion diffusion model (MDM) is a generative model that generates human motions. Given a text prompt describing the motion, the application would output a video showing a skeleton figure performing the action 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 significant number of timesteps in the reverse process, which means The current MDM does not satisfy the fast sampling in the learning trilemma (Figure 1).

1.2. Improving Sampling

Proposed by Xiao et al. (2022), the Denoising Diffusion GAN (DDGAN) is a novel generative model that utilizes GANs to satisfy all three requirements of the generative learning trilemma. Specifically, the usage of conditional GANs is what decreases the timesteps during sampling (Figure 2). As a result, training is a lot faster compared to traditional diffusion models. Furthermore, this enhancement prevents issues such as mode collapse and overfitting [3].

1.3. Integrating Motion Diffusion Model Into DDGAN

Initially, DDGAN was not designed for human motion data, but can be modified to do so. Specifically, DDGAN is built to handle image loss not motion or geometric loss which are complete different from each other. Therefore, we had to modify the DDGAN so that it can handle motion loss before integrating MDM to it. As for MDM, we needed to remove some existing logic in the current worklflow to make the final addition of the DDGAN (Section 3).

2. Related Work

Before we can discuss our work, we first need to discuss the foundations on which our work is built

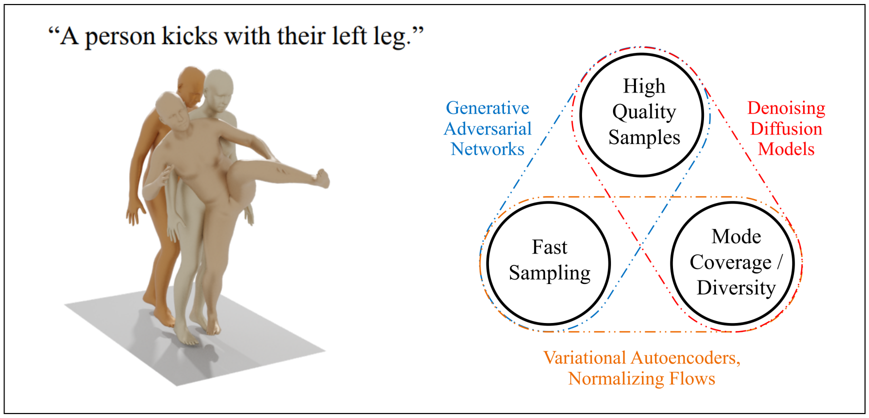
"A person kicks with their left leg."
High Quality Samples
Generative Adversarial Networks
Denoising Diffusion Models
Fast Sampling
Mode Coverage / Diversity
Variational Autoencoders, Normalizing Flows

Figure 1. Motion synthesis and the Generative learning trilemma [2] [3]

upon. Specifically we look at the Diffusion model, the Human Motion Diffusion Model, and finally the Denoising Diffusion GAN.

2.1. Diffusion

Denoising diffusion probabilistic models (DDPMs) [6,9] are a class of generative models that are able to generate high-quality images. These models use a diffusion process to iteratively transform a noise input into an output that approximates the target distribution. The diffusion process can be viewed as a continuous-time Markov chain that defines a sequence of conditional distributions that are used to approximate the target distribution. DDPMs have been shown to be effective in modeling complex high-dimensional data, such as natural images, and have achieved state-of-the-art performance on various image generation tasks. In the forward process, noise is added to the data according to a noise schedule β_t, in T steps:

$$q\left(\mathrm{x}_{1:T} \mid \mathrm{x}_0\right) = \prod_{t\geq 1} q\left(\mathrm{x}_t \mid \mathrm{x}_{t-1}\right) \tag{1}$$

$$q\left(\mathrm{x}_t \mid \mathrm{x}_{t-1}\right) = \mathcal{N}\left(\mathrm{x}_t; \sqrt{1-\beta_t}\mathrm{x}_{t-1}, \beta_t \mathrm{I}\right) \tag{2}$$

In the reverse process, the noise is removed from the data in the same order:

$$p_\theta\left(\mathrm{x}_{0:T}\right) = p\left(\mathrm{x}_T\right) \prod_{t\geq 1} p_\theta\left(\mathrm{x}_{t-1} \mid \mathrm{x}_t\right) \tag{3}$$

$$p_\theta\left(\mathrm{x}_{t-1} \mid \mathbf{x}_t\right) = \mathcal{N}\left(\mathrm{x}_{t-1}; \boldsymbol{\mu}_\theta\left(\mathrm{x}_t, t\right), \sigma_t^2 \mathrm{I}\right) \tag{4}$$

The main assumption on which this model is based is that the denoising distribution can be modeled by a Gaussian distribution and this assumption only holds when the denoising steps are large, in order of hundreds or thousands.

2.2. Denoising Diffusion GANs

The problem with having a large number of denoising steps is that the model is slow to sample from. Generations typically take minutes. As described previously the assumption of a normal guassian distribution causes the reverse process to create as many small steps as possible to approximate it. Ideally, a solution could come where instead of many small steps being used, a smaller amount of large denoising steps could be used. Unfortunately due to the assumption of a normal distribution this isn't possible with a diffusion model. This is where the Denoising Diffusion GAN (DDGAN) [3] comes in. The DDGAN is a generative model that utilizes GANs to satisfy all three requirements of the generative learning trilemma. It basically stipulates that sampling from pure noise is not efficient and proposes that sampling from a latent space is better. It does this by building off of an NCSN (noise-conditional score network) [10] [3]. In order to model the non-normal distribution that may result from an increased timestep size in the reverse process, a latent z variable is introduced which allows the modeling of various arbitrary distributions. This results in a multimodal denoising distribution which does not require a large number of denoising steps. The training process involves training a generator and a discriminator. The generator is trained to generate samples from the denoising distribution and the discriminator is trained to distinguish between samples from the denoising distribution and the data distribution. Adversarial loss is used to train the generator and discriminator.

2.3. Human Motion Diffusion Model

Realistic human motion generation is a challenging task due to the complex temporal structure of human motion data. The human motion diffusion model (HMDM) [2] is a generative model that is able to generate realistic human

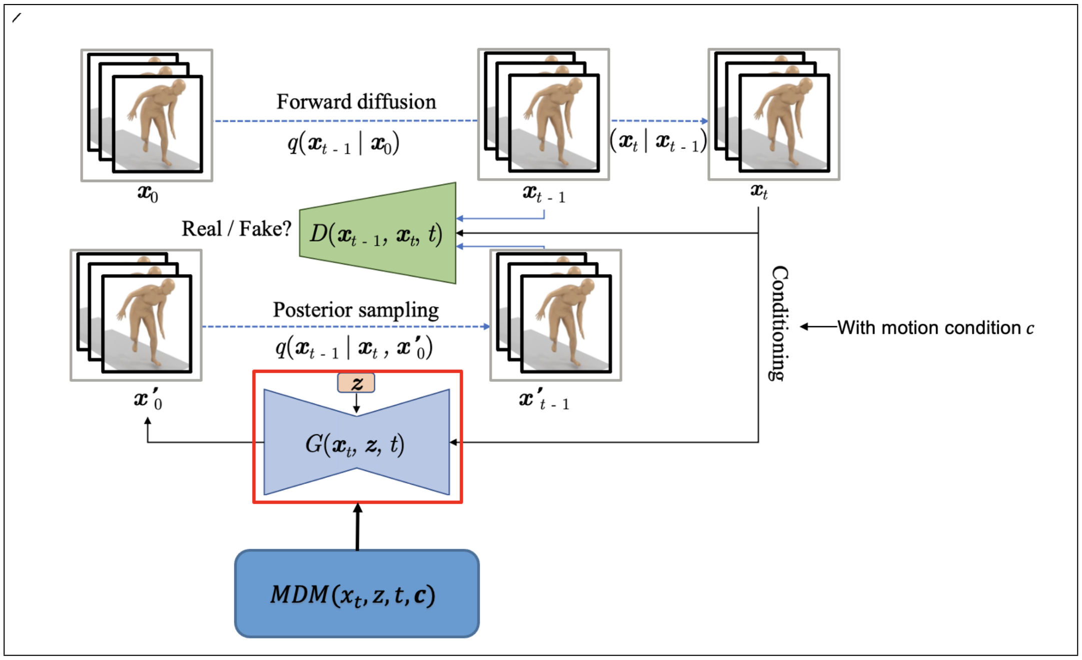
Forward diffusion
$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0)$
$(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$
$\boldsymbol{x}_0$
$\boldsymbol{x}_{t-1}$
$\boldsymbol{x}_t$
Real / Fake?
$D(\boldsymbol{x}_{t-1}, \boldsymbol{x}_t, t)$
Posterior sampling
$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}'_0)$
$\boldsymbol{x}'_0$
$\boldsymbol{x}'_{t-1}$
Conditioning
With motion condition c
$\boldsymbol{z}$
$G(\boldsymbol{x}_t, \boldsymbol{z}, t)$
$MDM(x_t, z, t, \boldsymbol{c})$

Figure 2. Denoising Diffusion GANs Architecture [3].

motion sequences. This was the first work to apply the diffusion model to human motion data. The model is conditioned using CLIP [8] based textual embeddings of the motion description. The noised motion sequence is fed to the transformer model which is trained to denoise the motion sequence. Since this model relies on conventional diffusion process, sampling from it is very slow. As we will see in the next section, we will be able to speed up the sampling process by almost a factor of 100 by integrating the MDM with the DDGAN.

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 [3]. 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 [3]. 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 [11]. The latent z variable is added to the other inputs described in [2]. 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.

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 [4]. Instead of using the generator provided by [3] 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.

3.1.3 DDGAN Sampling

In addition, to the modifications oriented towards the the training of the model, the reverse process needed to be refactored as well. Due to the original MDM reverse process assuming that is samples a denoising distribution from the normal distribution, it cannot be used as-is for a our modified architecture. Instead we adapted the sampling functions found in [2] and replaced any sampling from the MDM model to point to sample to the M2D model.

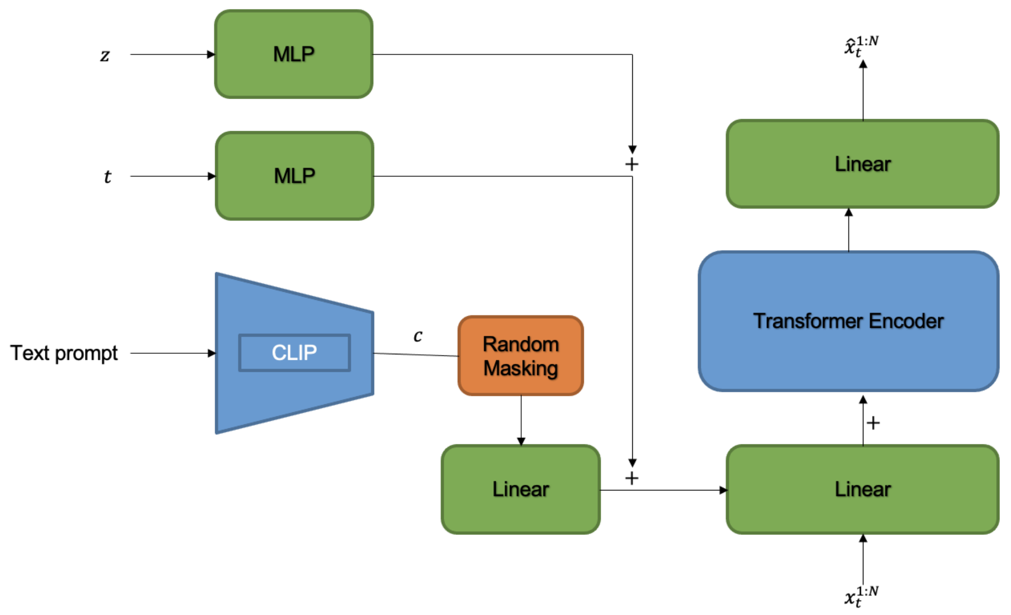

Figure 3. Our modified MDM architecture based on [2], we introduce the latent z variable by adding it with the timestep condition and the text prompt condition c

3.2. Adapting The Loss

As described in 3.1.2 we are using the discriminator to validate the values given from our generator. The discriminator offers us adversarial loss, however, this loss alone is not sufficient for our generator. Our generator being an MDM requires additional losses to produce high-quality outputs. For this we borrow the geometric losses described in [2]. Even though adversarial loss is not sufficient to train 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 5.

$$\mathcal{L}_{all} = \lambda_{adv}\mathcal{L}_{adv} + \lambda_{pos}\mathcal{L}_{pos} + \lambda_{vel}\mathcal{L}_{vel} + \lambda_{foot}\mathcal{L}_{foot} \tag{5}$$

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, where we distributed our learning across two physical nodes for a total of 4 GPUs. 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 [2], was employed in our text-to-motion generation task. HumanML3D [4] is a combination the HumanAct12 [5] and Amass [7] 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 [2] 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 [2]. Geometric losses are already represented in the HumanML3D ensemble, thus it is justifiable to omit them during training.

4.1. Unsuccessful Experiments

We performed several unsuccessful experiments in order to increase the quality of our generations, some of which we would like to highlight:

4.1.1 Smaller Discriminator

On observing that our discriminator consistently outsmarted the generator, we experimented with different architectures for our discriminator. One attempt was in using a discriminator that was not as complex. We used a discriminator with 4 downsampling blocks, instead of 6, and a final output of 128 channels, instead of 256. Doing this actually resulted in worse performance.

4.1.2 Discriminator with text condition

Another attempt was in conditioning our discriminator with text-encoding. The idea behind this was that having a similar architecture for both the generator and discriminator would benefit the generator. We found the opposite to be true, as this introduced undesirable bias in our discriminator.

4.1.3 Longer training

One suspicion regarding the worse accuracy of the model compared to MDM was possibly a lack of training. As such, it was proposed we train for as long as possible to observe if the model could still learn more features to yield better outputs. More specifically we trained the model on 2000 epochs which was far more extensive than our 350 epoch trials beforehand. As with the other experiments mentioned in this section, this did not improve our results and instead made our results far worse. Because of this testing we decided that 350 epochs would be the best happy-medium for our training purposes. As for why this occurs is most likely to do either with over-fitting and/or the generator failing. In second case we suspect the discriminator is near-perfect at detecting synthetic distributions, over time the generator keeps failing to the point where it outputs the same kind of results over and over which happen to be subpar.

4.1.4 Lowered Discriminator Learning Rate

Because our discrminator seemed to learn much faster than our generator and a smaller generator had already been shown to make performance worse. We instead opted to lower the learning rate of the discriminator. As we observed after testing, this was once again not successfull. The discrimator loss did reamin larger during the start of the training however over time it still converged too quickly to make a noticable impact on the end performance.

4.1.5 DDGAN sampling timestep modification

Another possibility that was discussed was maybe the generations weren't as accurate as the diffusion counterpart because we were not sampling from enough timesteps. Both a lower and higher sampling timestep was chosen however, both of these gave far worse generations. It seems that the reccomendation of 4 timesteps from [3] was very accurate and is also applicable for our purposes in the motion domain.

4.1.6 PIDM Integration

Lastly, we attempted to integrate PIDM with our motion generations. This is covered in more detail in Section 6.

4.2. Improvements

During our experimentation, we noticed that the discriminator exhibited a consistently lower loss in comparison to the generator. In order to address this imbalance, we took the proactive measure of reducing the learning rate of the discriminator. This strategic adjustment resulted in a more balanced distribution of losses between both the generator and discriminator, ultimately leading to a significant enhancement in the overall quality of our generated output. Through this process, we were able to achieve a more stable and reliable performance from our model. We discovered that by utilizing the pre-trained weights of the generator from the MDM model, which shared the same transformer encoder architecture as our own generator, we were able to achieve remarkable improvements in a notably shorter amount of time. This can be attributed to the fact that the MDM model was already trained on the extensive HumanML3D dataset.

5. Results

5.1. Qualitative Results

Our qualitative findings illustrate that our model is capable of most generations that MDM can perform. 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.

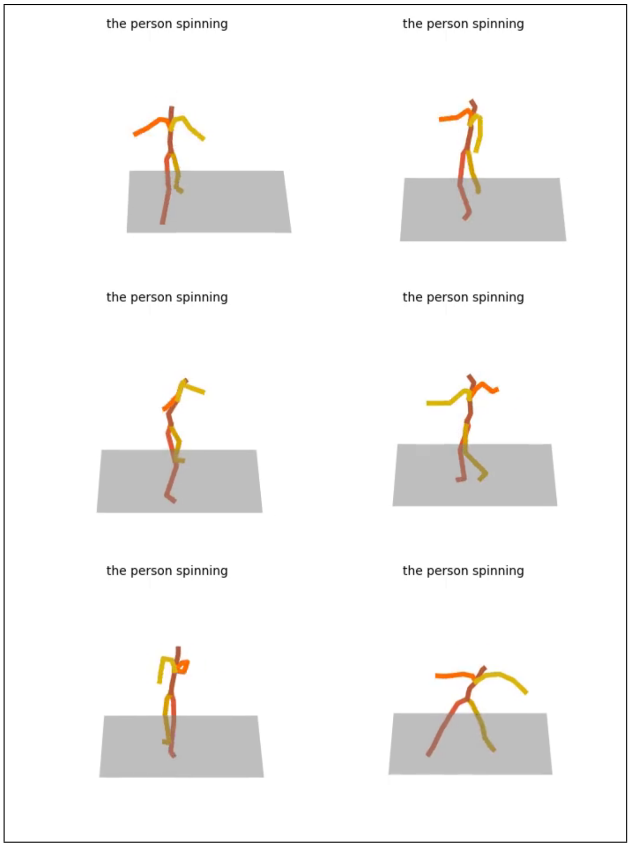

Figure 4. The shows a person in the act of spinning. We have have reduced the number of frames in order to fit the figure.

5.2. Limitations

Despite its realistic results, our model is not without its limitations. First of all, our model is only capable of generating motions up to 4 seconds in length which is short as compared to 9 second generations of MDM. After 4 seconds, the results from our model 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 also observed in the MDM generations but after 9 seconds. We believe this might be linked to the size of the latent space but we leave this for future work.

5.3. Quantitative Results

As a reminder, it is pertinent to recall the evaluation metrics employed in our model assessment. R-Precision and Multimodal-Distance scores measure the similarities between the generated motions and their respective text prompts. FID scores evaluate the similarity between the distributions of the generated motion and the real motion. Diversity quantifies the variability or dissimilarity among the generated motions. Finally, MultiModality refers to the mean variance observed during generation given a single text prompt.

We compare our results with the results of [2] and other motion generation models, which are shown in Table 1 and 2. We can observe that MDM-2-DiffGAN performs fairly well when compared to MDM. While we don't set the state-of-the-art results in any of the evaluation 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, which is likely because of the shorter generations of our model as discussed in the previous section. Despite its shortcomings, MDM-2-DiffGAN is able to generate motions around 100x faster than MDM.

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	$\mathbf{0.740}_{\pm.003}$	$1.067_{\pm.002}$	$\mathbf{3.340}_{\pm.008}$
MDM	$0.611_{\pm.007}$	$\mathbf{0.544}_{\pm.044}$	$5.566_{\pm.027}$
M2D (ours)	$0.698_{\pm.007}$	$2.44_{\pm.429}$	$6.353_{\pm.076}$

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	$\mathbf{9.559}_{\pm.086}$	$\mathbf{2.799}_{\pm.072}$
M2D (ours)	$9.416_{\pm.057}$	$2.671_{\pm.025}$

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

We perform sampling speed tests, where we measure the time it takes MDM-2-DiffGAN to generate motions. We execute 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 3, we can observe that it takes MDM considerably longer to generate motions than MDM-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

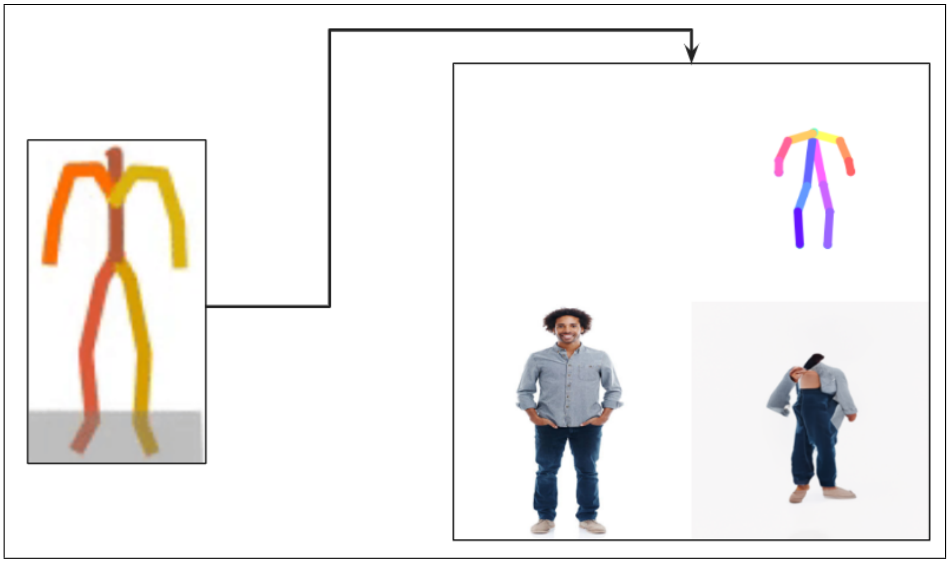

Figure 5. The target pose generated by our motion diffusion model was suppose to represent a person who is facing their back towards us.

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.

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

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

6. Additional Applications

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

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

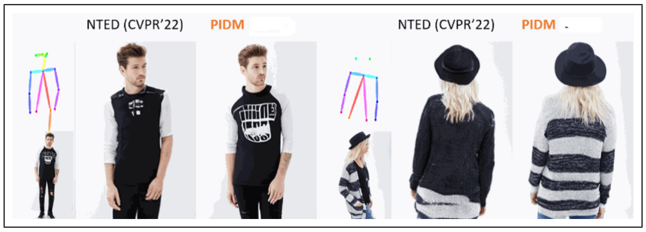

Figure 6. Image sample of the pose-guided image synthesis.

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. [1] utilizes a denoising diffusion model to generate the image samples (Figure 6).

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 7). The target pose is 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.

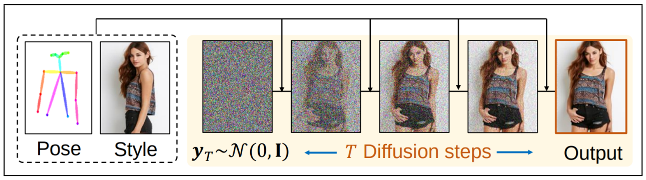

Figure 7. General workflow for the Person Image Synthesis Denoising Diffusion Model.

In Figure 4, 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. The first step was to determine a mapping from the motion diffusion skeleton to the correct color-codes for PIDM. Next, we wanted 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.

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 5). 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. Future work would be to find a way to generate a mapping from our skeleton to the correct pose features that the PIDM model expects.

7. Conclusions

In this work, we have presented MDM-2-DiffGAN, a hybrid model that combines the strengths of MDM and DDGAN. Although MDM has a limitation of long inference time, MDM-2-DiffGAN is capable of generating samples in significantly less time. As such, we have demonstrated that our method has superior scalability compared to MDM. Despite its merits, our approach does have some shortcomings, with the primary weakness being its low FID scores. At the time of writing this paper, we are still working on understanding the reason(s) for this. Our main suspicion is that it is possible that our plane shifting problem is correlated with our low FID scores. At a broader context there is most likely a plethora of areas in DDGAN that are still more tailored to image generation tasks instead of motion generation. In the future it's worth looking into these areas in far more detail to see how they can be refactored to be more motion-friendly. Addressing this issue has the potential to yield metrics that are comparable to those obtained in the original work. In the future, we would also like to expand the application of this approach to PIDM or a comparable model, with the aim of achieving comprehensive and realistic human motion generation.

References

[1] Ankan Kumar Bhunia et al. Person image synthesis via denoising diffusion model, 2023. 7

[2] Guy Tevet et al. Human motion diffusion model, 2023. 2, 3, 4, 6

[3] Zhisheng Xiao et al. Tackling the generative learning trilemma with denoising diffusion gans, 2022. 1, 2, 3, 5

[4] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5152–5161, 2022. 3, 4

[5] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2021–2029, 2020. 4

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. 2

[7] Naureen Mahmood, Nima Ghorbani, Nikolaus Troje, Gerard Pons-Moll, and Michael Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5442–5451, 2019. 4

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 3

[9] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. 2

[10] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020. 2

[11] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans, 2022. 3