DiffuseGaitNet: Improving Parkinson's Disease Gait Severity Assessment With a Diffusion Model Framework

Arshak Rezvani, Nasrin Ravansalar, Mohammad Ali Akhaee, Andrew J. Greenshaw, Russell Greiner, Maryam S. Mirian, Muhammad Yousefnezhad, and Martin J. McKeown

***Abstract*— Assessing the severity of gait impairment in Parkinson's disease (PD) using the Movement Disorder Society's Unified Parkinson's Disease Rating Scale (MDS-UPDRS) is typically performed by clinical experts, but this process is time-consuming, subjective, and costly. To address these challenges, we propose a Guided Diffusion Model with an encoder-only transformer that automatically predicts gait severity by learning the underlying distribution of PD gait and leveraging domain knowledge critical for clinical evaluations. Our diffusion model enables us to generate synthetic PD gait video frames conditioned on clinical features determined by experts to assess disease severity. These synthetic samples contain novel movement patterns not present in the observed data; systems trained on this information have better prediction performance. In addition, we propose a novel classification algorithm that can learn a predictive model, from both observed training data and synthetic samples, to accurately assess PD severity. We evaluate the effectiveness of the proposed method using two human motion datasets across two tasks: PD severity prediction and action classification. Our approach in predicting PD, and our action classification is sufficiently accurate that it can be applied to general applications with healthy subjects performing similar tasks. The full codebase is available on GitHub: https://github.com/arshakRz/DiffuseGaitNet**

***Index Terms*— Diffusion models, generative AI, transformers, attention-based networks, Parkinson's disease, gait impairments, human action recognition, MDS-UPDRS.**

I. Introduction

PARKINSON'S disease (PD) is a progressive neurodegenerative disorder where gait impairment is a significant motor symptom. Accurate assessment of gait severity is crucial for effective management and treatment. Traditionally, these assessments rely on the Movement Disorder Society's Unified Parkinson's Disease Rating Scale (MDS-UPDRS), which, while comprehensive, is time-consuming, costly, and subject to inter-rater variability. Advancements in AI offer the potential to automate and enhance these assessments [1]. However, existing methods often struggle to capture the complex nature of PD gait patterns and learning systems are limited by the availability of high-quality, labeled data. This paper describes an approach that learns the underlying distribution of variable gait patterns from PD data and uses it to learn a model that can accurately evaluate gait severity.

There are many reasons why learning the underlying distribution of PD gait patterns is useful; it is crucial for addressing several challenges in the assessment of Parkinson's Disease [2]. The variability in gait patterns among PD patients, coupled with the scarcity of publicly available data due to privacy concerns, makes it difficult to capture the full spectrum of symptom variations [3]. Without understanding this distribution, classification models are prone to overfitting on limited, labeled datasets, failing to generalize to new, unseen data. Additionally, class imbalances, where certain PD symptoms are underrepresented, further complicate the development of robust models. Privacy concerns also limit data sharing, hindering collaboration and validation efforts across institutions. Moreover, many existing classification models struggle to incorporate measures of uncertainty, reducing their reliability and generalizability in real-world clinical settings [1].

To overcome the challenges in assessing PD gait dysfunction, our work uses a generative approach to learn the underlying distribution of PD gait patterns. By accurately

Received 15 November 2024; revised 29 May 2025; accepted 5 July 2025. Date of publication 14 July 2025; date of current version 31 July 2025. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), Canada awarded to MJM and ZJW. *(Corresponding author: Martin J. McKeown.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the University of British Columbia's Human Ethics Board.

Arshak Rezvani, Maryam S. Mirian, and Martin J. McKeown are with the Department of Medicine, The University of British Columbia (UBC), Vancouver, BC V6T 1Z4, Canada (e-mail: arshak.rezvani0@gmail.com; maryam.mirian@ubc.ca; martin.mckeown@ubc.ca).

Nasrin Ravansalar and Mohammad Ali Akhaee are with the Electrical and Computer Engineering (ECE) Department, University of Tehran, Tehran 1417935840, Iran (e-mail: ravansalarn@ut.ac.ir; akhaee@ut.ac.ir).

Andrew J. Greenshaw is with the Department of Psychiatry, University of Alberta (UoA), Edmonton, AB T6G 2G5, Canada (e-mail: andy.greenshaw@ualberta.ca).

Russell Greiner is with the Department of Computing Science, UoA, Edmonton, AB T6G 2G5, Canada (e-mail: rgreiner@ualberta.ca).

Muhammad Yousefnezhad is with the Department of Computing Science and the Department of Psychiatry, UoA, Edmonton, AB T6G 2G5, Canada (e-mail: myousefnezhad@ualberta.ca).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNSRE.2025.3589074, provided by the authors.

Digital Object Identifier 10.1109/TNSRE.2025.3589074

capturing this distribution, we can better represent the complex and varied nature of PD symptoms, even with limited data availability. This method not only helps in creating high-fidelity representations of gait patterns but also enhances the performance of classification models both in a general movement perspective and in PD-specific scenario, enabling them to generalize better, manage class imbalances more effectively, and incorporate uncertainty measures. In this study, we propose an innovative method to predict gait severity in PD patients by leveraging domain knowledge critical for clinical evaluations. As described in more detail below, we employ a guided diffusion model with an encoder-only transformer as a noise predictive network, generating 30-frame windows of 15 joints in 3D space conditioned on features used by professionals to assess gait. By using this model, we can train a classifier using a leave-one-subject-out schema, allowing us to estimate the gait data distribution of the test subject and predict their gait score. This method addresses the limitations of traditional clinical assessments and current ML models, offering a robust, scalable solution for PD gait severity prediction. We evaluated the effectiveness of the proposed method using three human motion datasets across two tasks, demonstrating superior results in PD severity prediction compared to state-of-the-art methods, ultimately contributing to improved patient outcomes and more efficient clinical results.

The following sections detail: II. Related Work, III. Materials and Methods, IV. Experiments, and V. Discussion, demonstrating the potential of guided diffusion models in transforming PD gait assessment.

II. Related Works

Classic works on video-based assessment of PD symptoms have employed methods to manually design PD feature extraction functions and utilize machine learning (ML) techniques. Random forests [4], [5] and support vector machines (SVMs) [6] are among the most commonly used approaches in these classic PD assessment schemes. Some recent studies have moved towards deep learning (DL) methods to automate video based PD assessments, facilitating more detailed extraction and analysis of PD features [7], [8], [9], [10], [11], [12]. To predict the severity of PD from gait videos, authors [7] proposed a spatial-temporal attention graph convolutional network (2s-ST-AGCN). This model processes 2D predicted skeletons in two streams, *i.e.*, joints and bones, each through ten attention-aware ST-GCN units, which are then fused for severity prediction. Another approach, skeleton-silhouette fusion [8], extracts features based on skeletons and silhouettes from gait videos, using specialized convolutional layers. However, these techniques are prone to overfitting when dealing with small datasets [8].

To mitigate the risk of overfitting, one effective strategy is to design a more compact network, such as OF-DDNet [9], which builds on the Double-feature Double-motion Network (DD-Net) [13]. OF-DDNet processes 3D skeletons using temporal convolutional units and leverages ordinal-focal supervision. Since OF-DDNet is a deterministic model, it lacks measures of uncertainty, which limits its generalizability across different PD datasets [14]. Another strategy involves fusing different data types in the latent space and processing them through a probabilistic model. Incorporating expert knowledge -which is usually subjective and sometimes inconsistent between raters- from clinical concepts and class descriptions by vision-language models [10] or utilizing them in defining weak supervision labeling functions for PD classification tasks (WS-PD) [11] are as such examples. These strategies can reduce overfitting and address the lack of uncertainty issue, but challenges such as class imbalance due to small dataset sizes remain significant.

Generative models can help to measure uncertainty by learning the underlying data distributions [15]. These models can utilize gait videos involving human motions to extract gait features more effectively. Building on this idea, Gait Forecasting and impairment estimation transforMer (GaitForeMer) [12] has been proposed as a human motion forecasting model that uses an auto-encoder (AE) based on non-autoregressive transformers [16]. This model also estimates gait severity from the latent layer of the AE. To reduce the risk of overfitting, GaitForeMer is first pretrained on a large public action recognition dataset. However, the non-autoregressive design of the model can result in increased error accumulation, which may limit its effectiveness when dealing with long sequence data [12], [16].

While human motion prediction enhances the detailed extraction of gait features, human motion generation can also create new motions, helping to address challenges like small sample sizes and class imbalances [17]. A common strategy to guide generated motions is action conditioning [18]. For instance, Action2Motion [19] employs temporal variational AEs (VAEs) to generate class-conditional human motions at the frame level, while Action-Conditioned TransfORmer VAE (ACTOR) [17] does so at the sequence level using transformer VAEs. Action-conditioned motion transFormer (ActFormer) [20] combines the generative capabilities of GANs with transformers to produce more diverse and long-term sequences. However, action conditioning alone is insufficient for generating the stylized and controllable motions required for PD gait data synthesis. Diffusion models [21], known for their multi-step processing capabilities, excel in stylization and controllability in data generation. As an example of a related study, MotionDiffuse [22] incorporates text-driven embeddings into a diffusion model at each step to produce more diverse conditional motions. Below, we empirically compare our proposed method to some of these approaches.

III. Materials and Methods

The proposed method uses a guided diffusion model to analyze videos annotated with class labels, learning the underlying distribution of body-part joints position across frames, conditioned on domain-specific features. The method is divided into two primary phases: generator training and classifier training. As illustrated in Fig. 1, during the generator training phase (Fig. 1, left panel), video footage undergoes 3D pose estimation, generating sequences of body-part joint positions segmented into windows. Key gait features, identified through domain-specific knowledge, such as arm swing symmetry, step length, trunk rotation, step width, and hand

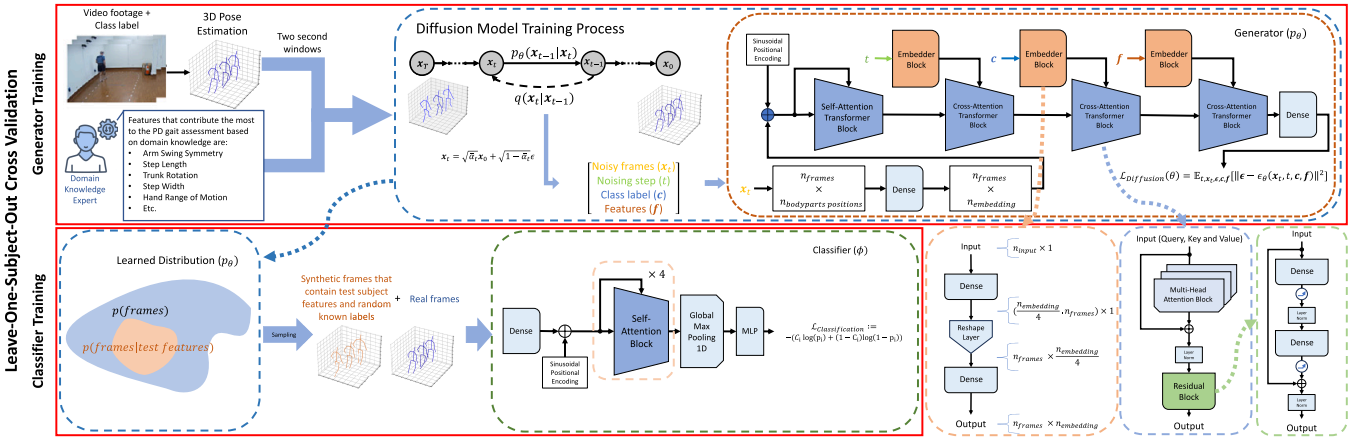


Fig. 1. The overall process. In the top left, video footage is used for 3D pose estimation, generating sequences of body-part joint positions. These joint positions are segmented into two-second windows, and domain-specific features used by clinicians to suggest a diagnosis of PD are extracted. In the top right panel, a diffusion model is trained to learn the distribution of PD gait, conditioned on these features and disease severity. The bottom left panel shows how this model generates samples mimicking the test set patterns. These samples are then added to the training set to train a classifier, which predicts the labels of the test set, as shown in the bottom middle panel.

range of motion—are extracted from these sequences. Then a diffusion model is trained to learn $p(\text{frames}|\text{class, features})$, in which frames are represented as 2D matrices of joint positions in $\mathbb{R}^{N_f \times N_j}$. Here, N_f denotes the number of frames in each window, and N_j represents the number of joints multiplied by three to account for each joint's 3D position. The window is chosen for two seconds due to the model's processing constraints, allowing efficient data handling while retaining essential gait features necessary for classification. These clips have demonstrated sufficient effectiveness in capturing prominent gait features, contributing to the model's overall classification accuracy without excessive computational demands.

After the model has learned the underlying distribution, as shown in Fig. 1 (bottom left), an accelerated sampling technique is employed in the classifier training phase to generate synthetic data. This synthetic data, which incorporates random labels and test set features, is combined with the original training set to form an augmented dataset. The final classifier is then trained on this enriched dataset, leveraging both real and synthetic data to improve its robustness. This classifier is subsequently used to predict the labels of the test set, ensuring a more reliable and generalized performance. By integrating domain-specific with advanced diffusion model techniques, our proposed method provides a robust framework for PD gait assessment, offering significant improvements in diagnostic accuracy and reliability.

A. Gait Generation

The denoising diffusion probabilistic model (DDPM) is a deep generative model designed to generate samples that mimic the underlying distribution of a dataset. The core idea is to train a denoiser that progressively improves the quality of noisy samples, making them increasingly similar to the original training data. If this process is done effectively, one can start with pure noise and iteratively refine it through multiple denoising steps until a sample that reflects the original distribution is produced. Here, we propose a customized derivation of this model tailored to effectively capture the complex dynamics of 3D motion data related to Parkinson's disease gait, with the added capability to condition the model on domain-specific features. A diffusion model consists of two primary processes: a forward (diffusion) process and a backward (denoising) process. In the forward process, noise is incrementally added to the data, creating a sequence of latent variables that form a Markov chain. This sequence eventually converges to a standard normal distribution, effectively obliterating the original data point's information. In the backward phase, the model is trained to recover the original data point from its noisy versions, essentially reversing the noise addition to reconstruct the original data [21].

As introduced before, let N_f denote the number of frames in each window, and N_j represent the number of joints multiplied by three to account for each joint's 3D position. The data matrix for a window of gait is expressed as $\mathbf{x}_t \in \mathbb{R}^{N_f \times N_j}$, where the index t indicates the position within the Markov chain. The distribution of the data at the tth position in the chain is represented by $q(\mathbf{x}_t)$. Let N_c denote the number of possible classes, and $\boldsymbol{c} \in \mathbb{R}^{N_c}$ be the corresponding one-hot encoded class label. Furthermore, let N_d denote the number of domain-specific features, and $\boldsymbol{f} \in \mathbb{R}^{N_d}$ be the associated feature vector.

The forward diffusion process initiates by sampling $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ from the original data distribution. At each subsequent step, small amplitude Gaussian noise is added, gradually transforming the sample into a noisier version. By step T (typically chosen as 1000) the sample $\mathbf{x}_T$ becomes nearly indistinguishable from a noise sample drawn from the isotropic Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, where $\boldsymbol{I}$ is the identity matrix with dimensions matching the data. Throughout this process, each intermediate sample, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{T-1}$, represents a progressively noisier transformation of the original data. This iterative addition of noise serves to transform the complex original data distribution into a tractable latent space.

This entire forward (diffusion) process, which transforms the intricate data distribution into a tractable latent space, is defined as:

$$q\left(\mathbf{x}_1, \ldots, \mathbf{x}_T \mid \mathbf{x}_0\right) := \prod_{t=1}^{T} q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right), \tag{1}$$

and the Gaussian transition between states is defined as:

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

where β_t values are scalars that control how much noise is added at each step, Although the values of $\beta_t \in (0, 1)$ could be learned, treating them as hyper-parameters reduces computational complexity without significantly impacting performance. Therefore, we define them using a fixed schedule. This schedule specifies the values of β_t for each step $t = 1, 2, \ldots, T$, ensuring that the βs are time-dependent and satisfy the condition $\beta_{t-1} < \beta_t$ for every $t > 1$.

Each intermediate latent state can be computed efficiently in closed form by defining $\alpha_t := 1 - \beta_t$ and $\overline{\alpha_t} := \prod_{s=0}^{t} \alpha_s$, as follows:

$$q\left(\mathbf{x}_t \mid \mathbf{x}_0\right) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\overline{\alpha_t}}\mathbf{x}_0, (1-\overline{\alpha_t})\boldsymbol{I}\right). \tag{3}$$

This allows any $\mathbf{x}_t$ to be expressed directly as a function of $\mathbf{x}_0$ and sampled noise ($\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$), bypassing the need to iterate through the Markov chain:

$$\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}) = \sqrt{\overline{\alpha_t}}\mathbf{x}_0 + \sqrt{1-\overline{\alpha_t}}\boldsymbol{\epsilon}. \tag{4}$$

This closed-form solution conserves computational resources by eliminating iterative loops for computing intermediate states.

The reverse (denoising) process is also a Markov chain, but with learned Gaussian transitions, starting from $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \boldsymbol{I})$ and defined as:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \tag{5}$$

With $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$ being neural networks, the Gaussian transitions defined as:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \tag{6}$$

To explain our choice of reverse process transitions, when β_t is sufficiently small (between 1×10^{-4} to 0.02), the reverse transition function closely resembles the Gaussian structure of the forward transition. This similarity is a key aspect of diffusion models, allowing deep neural networks to estimate these transitions [23], [24].

Now given the provided latents and their efficient computation, we can take the usual variational lower bound approach similar to VAEs to optimize the negative log likelihood, we can write it as:

$$\begin{aligned}\mathcal{L} &= \mathbb{E}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}] \\ &= \mathbb{E}_q[-\log p(\mathbf{x}_T) - \sum_{t \geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}{q(\mathbf{x}_t \mid \mathbf{x}_{t-1})}]\end{aligned} \tag{7}$$

To further simplify optimization, for the reverse process transitions $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$, we set $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) = \sigma_t^2 \boldsymbol{I}$ to untrained, time-dependent constants. Specifically, we set $\sigma_t^2 = \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$, which is the upper bound of this value, and the lower bound is $\sigma_t^2 = \beta_t$. While it is possible to interpolate $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$ between these bounds, our choice results in a MSE loss function instead of the ELBO. This occurs because the terms reduce to the KL divergence between Gaussian distributions p_θ and q. With fixed Gaussian variances, the KL divergence depends only on the difference of the means, leading to an MSE loss between the means of the distributions. Since q follows a Markov chain that adds noise to the data and we have a closed-form solution for it, we can reparametrize it as a function of the starting data point, the noising step, and some arbitrary noise. We can apply the same approach to p_θ, and with some simplification, derive a loss function that indicates we can predict the added noise at each step instead of the mean. The fine details of this derivation can be found in Appendix I. With this explanation, the simpler loss takes the following form, where $\boldsymbol{\epsilon}_\theta$, known as the noise predictive network, is a deep neural network that takes the noisy data point and noising step to predict the added noise at the given noising step:

$$\mathcal{L}_{\text{simple}}(\theta) = \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2\right]. \tag{8}$$

Conditional Generation: To generate data conditioned on both the class label and domain-knowledge features, we used Classifier-Free Guidance (CFG) [25]. This conditioning influences the generation process, steering it to produce samples that align with the specified class label and feature vector. The goal is to learn the target distribution given the class label c and the feature vector f, which can be expressed as follows:

$$q(\mathbf{x}_0 \mid \boldsymbol{c}, \boldsymbol{f}). \tag{9}$$

The CFG is a method to condition the generation process of the DDPMs that works by introducing the conditions to the noise predictive network by modifying it as:

$$\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \boldsymbol{c}, \boldsymbol{f}). \tag{10}$$

In earlier approaches, conditional generation was achieved using a pre-trained classifier that predicted the condition of the input sample. For example, when generating handwritten digits, the model could be conditioned to produce a specific digit by using a classifier trained to predict the digit label. This process involved incorporating the classifier's gradients with respect to the input to steer the generation. While effective, this method required a pre-trained classifier on the dataset. However, in our case, since the objective is to predict the labels, we cannot use this approach.

In contrast, in CFG, instead of training a separate classifier, we incorporate the conditions directly as inputs to the noise predictive network. This allows the network to learn to shape the output as intended to generate samples with the specified conditions. We train an unconditional DDPM, denoted as $p_\theta(\mathbf{x}_0)$, parameterized through a noise predictive network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$, alongside a conditional model $p_\theta(\mathbf{x}_0|\boldsymbol{c}, \boldsymbol{f})$, parameterized through $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \boldsymbol{c}, \boldsymbol{f})$. A single neural network is used to parameterize both models. For the unconditional model,

we simply input a null token $\emptyset$ as the condition identifier $\boldsymbol{c}$ when predicting the score, i.e., $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) = \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \boldsymbol{c} = \emptyset, \boldsymbol{f} = \emptyset)$. We jointly train both the unconditional and conditional models by randomly setting elements of c to the unconditional class identifier $\emptyset$ with a certain probability p_{uncond}, which is treated as a hyper-parameter. The sampling process is performed using a linear combination of conditional and unconditional score estimates. Here, $w \in \mathbb{R}$ is the coefficient that controls the ratio, known as the guidance weight, a hyper-parameter that balances the trade-off between diversity and fidelity in the generated samples. Higher values of w reduce variety while enhancing alignment with desired features. The sampling is described by the following equation:

$$\widetilde{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t, \boldsymbol{c}, \boldsymbol{f}) = (1+w)\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \boldsymbol{c}, \boldsymbol{f}) - w\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t). \quad (11)$$

The updated loss function with incorporation of conditions into learning the distribution is as follows:

$$\mathcal{L}_{\text{simple}}(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{c},\boldsymbol{f},\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \boldsymbol{c}, \boldsymbol{f})\|^2\right]. \quad (12)$$

Noise Predictive Network Architecture: The problem of training a conditional DDPM is reduced to training a neural network called the Noise Predictive Network, denoted as $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \boldsymbol{c}, \boldsymbol{f})$. This network takes the following inputs (Fig. 1, top middle): *Noisy frames* ($\mathbf{x}_t$), which represent the data at the current noising step t; *Noising step* (t) itself, a scalar indicating the current step in the diffusion process; *Domain-specific features* ($\boldsymbol{f}$), a vector of features that provide conditions for the generation process; and *Class* ($\boldsymbol{c}$), a one-hot encoded vector that represents the class label used as a condition for generation. The goal is for the network to predict the noise added to the data at each step, which is used to gradually reverse the diffusion process and generate samples conditioned on the given features and class.

To handle these inputs and capture the time-series nature of gait data, we implemented a transformer-based architecture to model the complex dynamics of 3D human motion (Fig. 1, top right). The network consists of transformer attention blocks. The first block uses self-attention to capture temporal dependencies within the joints positions, while subsequent blocks compute cross-attention to incorporate noising step and conditions of generation such as domain-specific features. These later blocks compute cross-attention between the embeddings of t, $\boldsymbol{f}$, and $\boldsymbol{c}$ and the output from the previous layer. This arrangement enables each layer to refine its focus on specific aspects of the data, incorporating external conditions to improve the model's performance.

Since $\mathbf{x}_t$ is a time series, a linear layer projects $\mathbf{x}_t$ at each time step. In contrast, t, $\boldsymbol{f}$, and $\boldsymbol{c}$ are static values without time dependencies; these are first projected into vectors using linear layers, then reshaped into matrices, and processed through an additional linear layer to match the shape of $\mathbf{x}_t$, making them compatible for use in transformer layers. To better capture the sequential nature of the data, sinusoidal positional embeddings are added to the projection of $\mathbf{x}_t$ to account for the time-series nature of the data, ensuring the model understands the order of the sequence. The final layer of the network is a linear projection layer that maps the embeddings back to the original joint position space. We employed weight tying between the linear layer for $\mathbf{x}_t$ and the output layer, using the same weight matrix enhances model optimization and reduce the parameter count. These layers do not include biases.

Accelerated Sampling: To sample from the learned distribution using a diffusion model, we employed the Denoising Diffusion Implicit Model (DDIM) sampler introduced by [26]. DDIMs accelerate the sampling process by utilizing Non-Markovian transitions instead of Markovian ones. This defines a generative procedure that allows sampling with fewer steps while maintaining the same training objectives as DDPMs. These transitions are designed to yield the same marginal distribution as DDPMs, which means we can optimize the same model for both. For detailed mathematical explanations, please refer to the original paper. They demonstrate that by selecting a subset of steps for generation, instead of all the steps in the Markov chain, we can achieve the same loss, and the terms not included in the subset are independent of neural network parameters. This approach effectively reduces computational load and significantly speeds up the sampling process. The reverse transitions in the DDIM sampler, using significantly fewer steps, are defined as follows:

$$\begin{aligned}\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\left(\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_\theta^{(t)}(\mathbf{x}_t)}{\sqrt{\alpha_t}}\right) \\ + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\boldsymbol{\epsilon}_\theta^{(t)}(\mathbf{x}_t) \\ + \sigma_t\boldsymbol{\epsilon}_t, \end{aligned} \quad (13)$$

where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \boldsymbol{I})$ is standard Gaussian noise independent of $\mathbf{x}_t$, and we define $\alpha_0 := 1$.

B. Classification

After training the Diffusion Model conditioned on features and labels, we generate a synthetic dataset with random labels from all classes and test set features using (13) (Fig. 1, bottom left) during the classifier training phase. In CFG (the framework for training conditional DDPMs), the hyper-parameter p_{uncond} controls how often the model receives null conditions, enabling it to learn from incomplete information. By randomly masking features and class labels with p_{uncond}, the model learns to generate data for various combinations of conditions, including those with missing information. Previous studies [27], [28], [29] have shown that Diffusion Models can effectively handle such independent conditions. Leveraging this, we independently mask features and labels with p_{uncond}, allowing the model to generate data based on any combination of conditions, including random class labels and test set features $p(\text{frames}|\text{class})$. The synthetic dataset, mimicking test set patterns, is added to the original training set (Fig. 1, bottom middle), and we then train a classifier on the augmented dataset. Algorithm 1 summarizes the proposed method.

Algorithm 1 Conditional Distribution-Based Classification

Input: Train set comprising datapoints (X_{train}), labels (y_{train}), and extracted features (f_{train}), and test set comprising datapoints (X_{test}) and extracted features (f_{test})

Parameter: Number of classes ($num_{classes}$) and Number of synthetic samples to generate ($n_{samples}$)

Output: Predictions for y_{test} using a classifier trained on synthetic and observed data

1: Learn conditional distribution (p_θ) of X_{train} with y_{train} and f_{train} as conditions based on (12).
2: Generate a synthetic labels y_{synt} by uniformly sampling $n_{samples}$ from the discrete set $\{1, 2, \ldots, \text{num}_{classes}\}$
3: Generate a synthetic dataset X_{syn} by sampling from the distribution p_θ conditioned on y_{syn} and f_{test} based on (13).
4: Train a classifier ϕ using $[X_{syn}, X_{train}]$ and $[y_{syn}, y_{train}]$.
5: Predicting test labels y_{test} using classifier ϕ from X_{test}

IV. Experiments

We conduct our experiments using three human motion datasets:

- **PD-Gait (A) [11]**: This dataset comprises 3D skeletal data from 15 key joints, collected from 29 individuals diagnosed with early-stage PD, with severity scores of 0 (slight) and 1 (moderate). Participants were filmed walking along an oval path for approximately 4 minutes, with videos recorded at 15 frames per second (fps). Each participant was assessed by a certified clinical research assistant using the Movement Disorder Society-sponsored Unified Parkinson's Disease Rating Scale (MDS-UPDRS). Due to ethical concerns related to patient privacy, the dataset is not publicly available.
- **HumanAct12 (B) [19]**: This publicly available dataset comprises 3D skeletal data with 24 joints, spanning 1,191 motion clips that are categorized into 12 distinct action types.
- **UESTC RGB-D (C) [30], [31]**: This dataset contains 25,600 video clips of varying-view action data, featuring 40 classes of aerobic exercises. We utilized the 3D skeletal data from 25 joints across all eight different views.

We used dataset A for predicting PD severity, while the other two datasets (healthy subjects) were utilized for action classification. Additionally, all datasets were employed for the motion generation task.

To evaluate the proposed method against state-of-the-art approaches, we considered three external benchmarks as well as an internal baseline derived from the DiffuseGaitNet framework. This baseline involves training the classifier directly on real gait data without any diffusion-based data augmentation—i.e., an end-to-end classification model using only the raw input data and the same architecture as used in our method. It serves to isolate the contribution of the synthetic data generation and conditioning modules in DiffuseGaitNet, enabling a controlled comparison.

The baseline was evaluated on both PD and action classification tasks. For external comparisons, WS-PD [11] serves as a benchmark for PD severity estimation. It employs a simple MLP trained on 3D poses using a weak supervision structure, where labels are derived from predefined PD gait features. As these labeling functions are dataset-specific, WS-PD is only applicable to the PD-Gait dataset. GaitForeMer [12] was used as another benchmark for both PD and action classification. We pre-trained this model on the NTU-RGB+D dataset [32], which contains 3D skeletal data for human action recognition, and then fine-tuned it on datasets A, B, and C.

To assess the motion generation capabilities of the proposed method, we compared it with ActFormer [20] across all three datasets, as ActFormer represents a state-of-the-art approach for motion generation. We conducted the comparison of synthetic and test data based on scores obtained from an ST-GCN [33]. To standardize the experimental conditions, we extracted 30-frame clips from all videos using a sliding window approach and selected the same 15 key joints from each clip. For the PD-Gait (A) dataset, we also extracted 15 features from the 3D human poses relevant to Parkinson's disease gait assessment—such as step length, foot clearance, hip and knee flexion, and step width—following the approach used in WS-PD [11]. These features were used in both the corresponding experiment and the proposed method. For the other datasets, we extracted nine features that capture body part dynamics and are pertinent to action classification.

For the experiments on dataset A, we applied a leave-one-subject-out cross-validation (LOSO-CV) approach at the window level, following the methodology outlined in [11]. In each iteration, one subject's data windows were set aside for testing, while the model was trained on the windows of the remaining 28 subjects. This LOSO-CV setup was chosen to ensure robust evaluation across subjects and reduce the risk of overfitting to specific individuals.

We reported accuracy, precision, recall, and F1 scores as the evaluation metrics, presented as $m \pm \text{SE}\%$. Here, m represents the average metric calculated at the subject level. To obtain m, we first classify each window for the test subject, then use majority voting to aggregate these window-level classifications into a single label for the subject. The metric (e.g., accuracy or F1 score) is then calculated based on this aggregated subject-level label, providing a measure of overall performance for each test subject. The standard error (SE) quantifies the variability in the metric across all test subjects. For each subject, we calculate the metric at the window level by evaluating the model's predictions for all windows belonging to that subject (e.g., if a subject has 1,000 windows and the F1 score for these predictions is 0.85, this score represents the window-level performance for that subject). We then compute SE by taking the standard deviation of these window-level metric values across all subjects and dividing it by the square root of the number of subjects, $\sqrt{N}$. This provides a measure of the consistency of the metric m across the dataset, with a smaller SE indicating more reliable performance across subjects.

For the action classification task, we employed 5-fold cross-validation, reporting accuracy as $m \pm \text{SE}\%$, where m is the mean and SE represents the standard error percentage, averaged across all folds.

Experiment Design: We trained a DDPM for each dataset, completing approximately 5k/3k/2k epochs per fold for datasets A/B/C, using a linear noise schedule over 1000 steps for βs with values ranging from 1×10^{-4} to 0.02. For the Noise Predictive Network ($\boldsymbol{\epsilon}_\theta(x_t, t, c, f)$), we employed mixed float16 precision, with the unconditional probability p_{uncond} fixed at 0.2.

The network architecture included 4/5/4 transformer attention blocks each containing 128/512/256 attention heads for datasets A/B/C, respectively. The dimensions for queries (q), keys (k), and values (v) were all set to 16, balancing computational efficiency and model performance. The embedding dimension was configured to 128/512/512 for datasets A/B/C. All linear layers within the transformer blocks, including those in the embedders and the dense layers of the MLPs, utilized a dropout rate of 0.1 and the L2 regularization factor of 0.001. The weight-tied input and output projection layers did not use dropout but instead applied a higher L2 regularization factor of 0.01 to enhance stability and prevent overfitting in critical input/output mappings. The ReLU activation function was used throughout the model, and layer normalization with $\epsilon = 1 \times 10^{-6}$ was applied within the model layers.

The models were optimized using the AdamW optimizer, selected for its effective weight decay mechanism that reduces overfitting, particularly beneficial for larger datasets. A cosine learning rate schedule was applied, with initial learning rates of $1e-3/1e-4/1e-4$ for datasets A/B/C, respectively.

Training was conducted on a system running Ubuntu 22.04.3 LTS, with TensorFlow 2.15.0, and the NVIDIA driver version 535.154.05. Training deep learning models — especially diffusion models and Transformer-based classifiers — is inherently resource-intensive. In our setup, the complete pipeline required approximately 10-12 GPU hours per subject using the two NVIDIA TITAN RTX GPUs. However, this computational demand is limited to the training phase.

At inference time, the resource requirements are substantially reduced. By employing key-value (KV) caching during Transformer inference, memory usage and latency are significantly lowered. This optimization enables the model to run efficiently on standard clinical hardware, including mid-range GPUs or high-performance CPUs, making the framework viable for deployment in real-world clinical environments with limited computational capacity.

For each fold, we generated 400/128/128 batches of 256/128/128 data points for datasets A/B/C, respectively, using 50 sampling steps and a guidance strength (w) of 0.9. The generated data was then concatenated with the fold's training data and shuffled. For training the classifier, we used a transformer-based architecture with the same parameters as the generator. The data was processed through a linear layer to obtain the embedding size, followed by sinusoidal positional encoding. The architecture comprised four transformer layers, followed by global max pooling to reduce the data from a time series to a vector, with a simple MLP performing the final classification task.

A. Results

We evaluated the proposed method on three datasets: PD-Gait, HumanAct12, and UESTC, comparing the performance against state-of-the-art models, including WS-PD, GaitForeMer, ActFormer, and a baseline (classifier trained on real data). The results are summarized in Fig. 2 and Fig. 3.

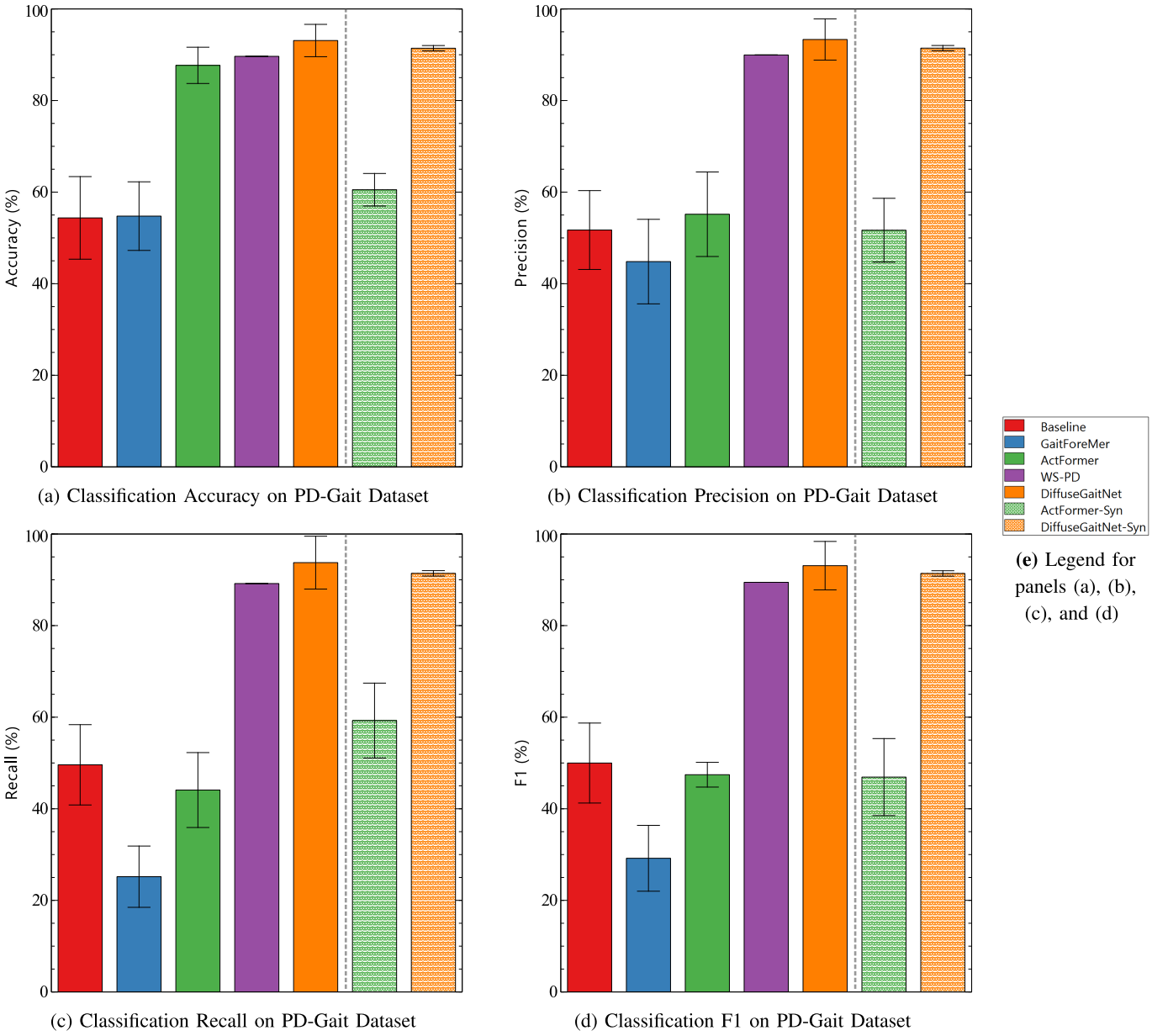
On the PD-Gait dataset, the proposed method achieved superior results, significantly outperforming the benchmarks across all metrics, including accuracy, precision, recall, and F1 score. Compared to WS-PD, which uses weak supervision, our proposed method provides a more robust and detailed prediction of PD severity, further confirming the effectiveness of incorporating domain-specific features in the generative process.

On the HumanAct12 dataset, which focuses on action classification in healthy subjects, our proposed method showed strong performance, achieving an accuracy of 85.36. The synthetic data generated during training provided diverse examples of actions, enhancing the classifier's ability to generalize across a variety of movement types. The HumanAct12 dataset, though simpler than PD-Gait in terms of task complexity and clinical relevance, demonstrated the model's consistency in generalization across real and synthetic data. The inclusion of domain-specific features, such as body part dynamics, enabled the model to better represent motion data, improving classification accuracy compared to methods like ActFormer.

The UESTC dataset presented significant challenges for both the proposed method and prior state-of-the-art models like ACTOR [17]. The proposed model achieved the accuracy of 61.99, and while this is lower than other datasets, it is worth noting that the authors of ACTOR [17] reported similarly low accuracy on UESTC (42.4) despite their sequence-level approach. This indicates that the low accuracy is likely due to the inherent complexity of the dataset rather than a shortcoming of our model alone. UESTC consists of aerobic exercises captured from multiple views, and these sequences require a broader temporal understanding of movements. While ACTOR's [17] sequence-level approach is generally more suited for long-term actions, their model also struggled on UESTC due to the difficulty in representing the complex rotational and positional data. Similarly, our proposed method, which uses a 30-frame window focusing on 3D joint positions, faces challenges in capturing long-term dependencies across multiple views, leading to the observed drop in accuracy. For datasets like UESTC, where long-term dependencies are crucial, synthetic data alone could not fully overcome the limitations posed by the complexity of the dataset. While synthetic data enhanced generalization to some extent, the absence of a sequence-level modeling approach remains a challenge for datasets involving more complex, long-term actions.

B. Ablation Studies

We conducted a comprehensive ablation study to evaluate the contribution of different components in *DiffuseGaitNet*,



(a) Classification Accuracy on PD-Gait Dataset

(b) Classification Precision on PD-Gait Dataset

(c) Classification Recall on PD-Gait Dataset

(d) Classification F1 on PD-Gait Dataset

(e) Legend for panels (a), (b), (c), and (d)

Fig. 2. Quantitative results of the proposed method, a baseline, and three benchmarks tested on the PD-Gait dataset (A). All models are trained using the LOSO-CV scheme. Results are presented as $m \pm$ SE%, where SE $= 0.00$ for WS-PD due to its unique weakly supervised design, which produces deterministic labels with no SE. Models labeled with "-Syn" indicate synthetic data generated by the corresponding method.

including the effect of synthetic data, domain-specific conditioning in the diffusion model, and the choice of classifier.

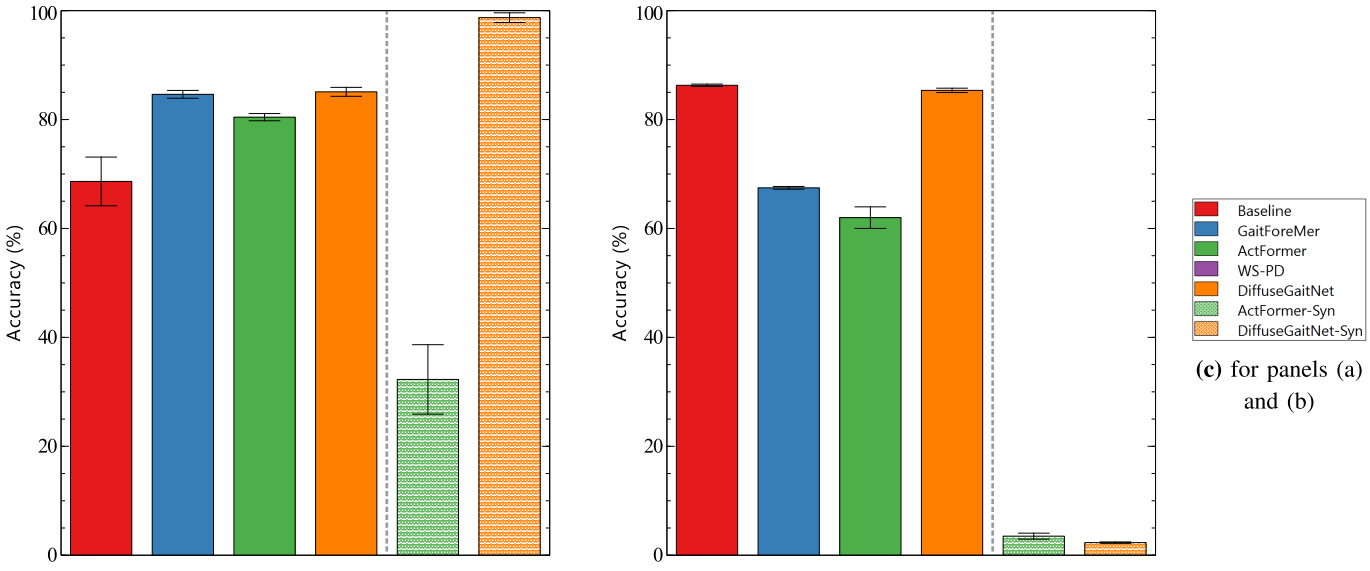
a) Effect of Synthetic Data and Conditioning: To assess the impact of synthetic gait sequences, we compared three settings: (1) training with real data only, (2) using additional synthetic data generated without conditioning, and (3) using synthetic data generated with domain-specific conditioning on clinical features (e.g., step length, trunk rotation, arm swing symmetry). The results demonstrate that conditioning significantly enhances the quality of generated sequences and leads to superior classification performance.

b) Effect of Classifier Architecture: We also compared two classifiers: a standard XGBoost model and the proposed Transformer-based architecture. While both classifiers benefited from the addition of synthetic data, the Transformer consistently outperformed XGBoost, especially when trained with conditionally generated sequences. This highlights the strength of the proposed architecture in modeling complex temporal and structural patterns in gait data.

These results confirm that both domain-specific conditioning and the choice of classifier significantly impact the overall performance of the system.

V. Discussion

The discussion section highlights the key contributions and implications of our guided diffusion model for PD gait severity assessment. The proposed method not only demonstrates superior performance compared to existing methods but also leverages synthetic data generation and 3D avatar visualization



(a) Classification Accuracy on HumanAct12 Dataset

(b) Classification Accuracy on UESTC Dataset

(c) for panels (a) and (b)

Fig. 3. Quantitative results of the proposed method, a baseline, and three benchmarks tested on the HumanAct12 dataset (B) and the UESTC dataset (C). All models are trained using a 5-fold cross-validation scheme. Results are presented as $m \pm \text{SE}\%$. Models labeled with "-Syn" indicate synthetic data generated by the corresponding method.

to enhance clinical utility. By addressing ethical considerations and improving the interpretability of predictions, our proposed method represents a significant advancement in the automated evaluation of PD gait, offering both practical and clinical benefits. Below, we discuss the clinical applications, comparative performance, impact of synthetic data, advantages of the generative model, and relevant ethical considerations.

A. Clinical Applications

The proposed DiffuseGaitNet model has significant potential for real-world clinical applications, particularly in neurological assessment and rehabilitation. One of the primary advantages of this approach is its ability to automate Parkinson's Disease (PD) gait severity assessment, reducing reliance on subjective clinical evaluations such as MDS-UPDRS, which are time-consuming and prone to inter-rater variability. By leveraging synthetic gait samples alongside real patient data, our model can provide objective, consistent, and scalable gait assessments that may support telemedicine applications and remote patient monitoring. This capability is particularly valuable in underserved or remote areas, where access to movement disorder specialists is limited. Additionally, DiffuseGaitNet can serve as an assistive tool in clinical decision-making, helping neurologists track disease progression and optimize treatment strategies. By providing a quantitative and automated gait assessment, this model could enhance longitudinal monitoring of PD patients, allowing for more precise tracking of symptom fluctuations in response to medication or therapy.

B. Comparative Performance

To evaluate the generalizability of our approach beyond Parkinsonian gait, we applied our model to two additional human motion datasets—HumanAct12 and UESTC—in addition to the PD severity prediction task. Across all datasets, the proposed method consistently outperformed or matched state-of-the-art models, demonstrating its effectiveness for both clinical and general human action recognition tasks. On PD-Gait, we outperformed the weakly supervised WS-PD model and the self-supervised GaitForeMer in both accuracy and precision. On HumanAct12, our proposed method performed competitively, matching or slightly surpassing the performance of ActFormer in terms of classification accuracy. The diffusion model's ability to model uncertainty and generate realistic samples provided a clear advantage in tasks where real-world data was limited. Although the proposed method underperformed on UESTC compared to datasets like PD-Gait and HumanAct12, the low accuracy is in line with what was observed by the authors of ACTOR [17], who also faced difficulties with this dataset. This suggests that UESTC poses significant challenges due to its inherent complexity, and both sequence-level and frame-level approaches struggle to fully capture the nuances of the data.

TABLE I
ABLATION RESULTS SHOWING THE EFFECT OF SYNTHETIC DATA, CONDITIONING, AND CLASSIFIER ARCHITECTURE

Classifier	Training Data	Conditioning	Accuracy (%)
XGBoost	Real only	—	51.72
XGBoost	Real + Synthetic	Unconditioned	55.17
XGBoost	Real + Synthetic	Conditioned	**58.62**
Transformer	Real only	—	54.37
Transformer	Real + Synthetic	Unconditioned	55.17
Transformer	Real + Synthetic	Conditioned	**93.10**

C. Impact of Synthetic Data

One of the key innovations of our proposed method is the generation of domain-knowledge-guided synthetic gait data to supplement the limited real-world datasets available for

PD. The scarcity of large, labeled clinical datasets poses a major challenge, often resulting in overfitting and poor generalization. To address this, we leverage a guided diffusion model to synthesize gait sequences conditioned on clinically relevant features such as step length, arm swing symmetry, trunk rotation, and stride variability. This approach enhances the training process through two complementary mechanisms. First, by incorporating expert-defined biomechanical patterns of Parkinsonian gait into the generated samples, the classifier learns to associate these meaningful variations with symptom severity more effectively. Second, the synthetic data enriches the feature space by introducing previously unseen but biomechanically plausible gait variations, helping to overcome class imbalance—particularly for underrepresented severity levels—and enabling the model to generalize better to unseen subjects. As a result, our classifier trained on this augmented dataset demonstrates enhanced robustness and predictive accuracy, as reflected in improved performance across held-out test sets and multiple datasets. Beyond improving model performance, the synthetic data also offers new insights into subtle movement patterns associated with PD that may not be captured in limited real data, with potential implications for both research and clinical interpretation. Looking forward, we plan to refine the conditioning strategies and investigate feature weighting schemes to further enhance both model interpretability and clinical utility. Extending the model to assess a broader spectrum of gait-related motor impairments could also support deeper understanding of disease progression and therapeutic response.

Furthermore, we collaborated with a movement disorder neurologist to qualitatively assess the realism of the synthetic gait sequences. A series of 2-second stick-figure animations—comprising both real and synthetic data from PD and healthy control (HC) subjects—were presented in random order without labels. The clinician was asked to identify whether each sample represented a PD or HC gait and provide a brief justification. The neurologist was able to correctly identify many PD vs. HC samples and noted that the synthetic PD samples exhibited hallmark gait abnormalities such as reduced stride length and decreased arm swing, while synthetic HC samples showed smooth, coordinated movement. This suggests that our model successfully captures meaningful and recognizable gait features relevant to Parkinsonian motor impairment. Refer to the supplementary material section for viewing the 2-second synthetic gaits evaluated by a movement disorder specialist.

D. Generative Model Advantages

The use of a guided diffusion model in this study offers several advantages over traditional generative approaches for analyzing PD gait. Unlike models that rely solely on observed data, our diffusion model can generate novel gait sequences that are realistic and clinically relevant. This capability is crucial for understanding the full spectrum of gait impairments in PD, which often exhibit subtle variations that are difficult to capture with standard methods. Furthermore, the multi-step processing nature of diffusion models allows for more precise control over the generated data, enabling the creation of stylized motions that closely align with clinical observations. A particularly innovative aspect of the proposed method is the creation of 3D representations of avatars based on the generated gait sequences. These avatars provide a visual and interactive representation of a patient's gait, allowing neurologists to better imagine and understand the patient's condition. This visualization aids in bridging the gap between quantitative model predictions and clinical intuition, enabling more informed and nuanced clinical assessments. The ability to visualize patient-specific gait patterns in 3D helps neurologists to identify subtle changes and variations in movement, which could be crucial for early detection and personalized treatment planning in PD. The enhanced controllability and fidelity of the generated data, combined with the 3D avatar visualization, distinguish the proposed method from previous methods, making it a valuable tool for both predictive modeling and exploratory analysis in neurodegenerative diseases.

E. Ethical and Practical Considerations

The integration of synthetic data generation into clinical assessment frameworks raises important ethical and practical considerations. While the use of synthetic data mitigates privacy concerns associated with sharing real patient data, it also introduces questions about the authenticity and reliability of the generated sequences. Ensuring that synthetic data accurately reflects the clinical reality of PD gait is paramount, as any deviation could lead to incorrect diagnoses or treatment plans. To address these concerns, our model incorporates domain-knowledge features explicitly derived from clinical expertise, ensuring that the synthetic data remains grounded in real-world clinical practices. Additionally, the diffusion model's ability to manage uncertainty provides a safeguard against over-reliance on synthetic data, enhancing the overall reliability of predictions. Practically, the adoption of such models could facilitate broader data-sharing initiatives across institutions, enabling more collaborative research without compromising patient confidentiality. As these technologies advance, it will be crucial to establish clear guidelines and standards for the ethical use of synthetic data in clinical decision-making. One last issue is about the chosen window length. While our diffusion model successfully utilizes 2-second clips, we acknowledge that clinical assessments typically prefer longer video sequences. Extended video clips enable clinicians to observe a broader range of movement patterns and detect subtle anomalies in gait that may not appear in shorter clips. For PD specifically, extended sequences provide more comprehensive information on movement regularity, step consistency, and trunk sway over time. In future work, we aim to expand the model's applicability by incorporating longer video clips, such as 10-second sequences, to capture a more representative sample of patient gait. Testing with extended clips would not only better meet clinical expectations but could also improve the model's robustness and generalizability across varied movement patterns.

VI. Conclusion

In this work, we introduced DiffuseGaitNet, a guided diffusion model framework for improving Parkinson's disease gait

severity assessment. By conditioning on clinically meaningful gait features and generating synthetic but biomechanically plausible sequences, our model enhances classification performance while maintaining clinical interpretability. We demonstrated that synthetic data improves generalization, mitigates class imbalance, and expands feature diversity beyond the limitations of real-world datasets. Future work will focus on clinician-in-the-loop refinement using 3-D gait visualizations and reinforcement learning with human feedback, aiming to align generative outputs more closely with neurologists' diagnostic criteria and further bridge the gap between AI-driven modeling and clinical decision-making.

Acknowledgment

The authors thank Dr. Taomian Mi for her careful evaluation of their generated gait visualizations and for providing clinical insights that helped validate the realism and interpretability of their synthetic samples.

Appendix I
Mathematical Details of DDPM [21]

The loss function defined in 7 can be further simplified to this three term loss:

$$\mathbb{E}_q\Bigg[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T \mid \mathbf{x}_0)\|p(\mathbf{x}_T))}_{L_T} + \sum_{t>1}\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)\|p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t))}_{L_{t-1}} + \underbrace{-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)}_{L_0}\Bigg] \tag{14}$$

Equation (14) uses KL divergence to directly compare $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ against forward process posteriors, which are tractable when conditioned on $\mathbf{x}_0$:

$$\begin{aligned} q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) &= \mathcal{N}\left(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right), \tilde{\beta}_t \boldsymbol{I}\right) \\ \text{where} \quad \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right) &= \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t}\mathbf{x}_t \\ \text{and} \quad \tilde{\beta}_t &= \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \end{aligned} \tag{15}$$

The loss function in (14) consists of three terms. The first term, L_T, is constant during training and can be ignored because the variances β_t in the forward process are fixed to constants, making the approximate posterior q non-learnable. The second term, L_{t-1}, represents the KL divergence between two Gaussian distributions at different noising steps. Since we have fixed the variances with $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_t^2 \boldsymbol{I})$, we can express it as follows:

$$L_{t-1} = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\left\|\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right) - \boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right)\right\|^2\right] + C \tag{16}$$

where C is a constant that does not depend on θ. Thus, we observe that the simplest parameterization of $\boldsymbol{\mu}_\theta$ is a model that predicts $\tilde{\boldsymbol{\mu}}_t$, the posterior mean of the forward process. However, we can use the closed-form solution of (4) and write $\mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon})$ and replace it in (16) to get:

$$\begin{aligned} &L_{t-1} - C \\ &= \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{2\sigma_t^2}\left\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon})) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)\right\|^2\right] \end{aligned} \tag{17}$$

Then we can replace $\tilde{\boldsymbol{\mu}}_t$ with applying the forward process posterior formula (15) and get:

$$\begin{aligned} &L_{t-1} - C \\ &= \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{2\sigma_t^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)\right\|^2\right] \end{aligned} \tag{18}$$

Equation (18) reveals that $\boldsymbol{\mu}_\theta$ must predict $\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right)$ given $\mathbf{x}_t$. Since $\mathbf{x}_t$ is available as input to the model, we may also reparameterize $\boldsymbol{\mu}_\theta$ as:

$$\boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)\right) \tag{19}$$

By plugging (19) into (18) we can derive this loss:

$$\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t\left(1-\bar{\alpha}_t\right)}\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2\right] \tag{20}$$

It is beneficial to ignore the t-dependent coefficients that are multiplied by the MSE loss. This is because frequency analysis shows that optimizing the loss for different t values is akin to optimizing the generation of details across various frequencies. Therefore, we can treat the detail generation of different frequencies as equally important by disregarding the coefficients.

For the last term, L_0, we can approximate it as Gaussian with learned parameters, similar to the reconstruction term in VAEs [34]. In this context, the negative log-likelihood of a Gaussian distribution simplifies to a MSE, up to a constant factor. Additionally, we can ignore the coefficients for this term since they do not affect the optimization, as they are constants that do not influence the gradients or reltive scaling of other terms in the loss function.

All of this allows us to express a simplified loss function for training DDPMs, the same as (8), as follows:

$$\mathcal{L}_{\text{simple}}\left(\theta\right) = \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)\right\|^2\right]. \tag{21}$$

References

[1] K. W. Park, M. S. Mirian, and M. J. McKeown, "Artificial intelligence-based video monitoring of movement disorders in the elderly: A review on current and future landscapes," *Singap. Med. J.*, vol. 65, no. 3, pp. 141–149, 2024.

[2] Y. Guo, J. Yang, Y. Liu, X. Chen, and G.-Z. Yang, "Detection and assessment of Parkinson's disease based on gait analysis: A survey," *Frontiers Aging Neurosci.*, vol. 14, Aug. 2022, Art. no. 916971.

[3] R. Ranjan, D. Ahmedt-Aristizabal, M. Ali Armin, and J. Kim, "Computer vision for clinical gait analysis: A gait abnormality video dataset," 2024, *arXiv:2407.04190*.

[4] M. H. Li, T. A. Mestre, S. H. Fox, and B. Taati, "Vision-based assessment of parkinsonism and levodopa-induced dyskinesia with pose estimation," *J. NeuroEng. Rehabil.*, vol. 15, no. 1, pp. 1–13, Dec. 2018.

[5] G. Sarapata et al., "Video-based activity recognition for automated motor assessment of Parkinson's disease," *IEEE J. Biomed. Health Informat.*, vol. 27, no. 10, pp. 5032–5041, Oct. 2023.

[6] Y. Liu et al., "Vision-based method for automatic quantification of parkinsonian bradykinesia," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 10, pp. 1952–1961, Oct. 2019.

[7] R. Guo, X. Shao, C. Zhang, and X. Qian, "Multi-scale sparse graph convolutional network for the assessment of Parkinsonian gait," *IEEE Trans. Multimedia*, vol. 24, pp. 1583–1594, 2022.

[8] Q. Zeng, P. Liu, N. Yu, J. Wu, W. Huo, and J. Han, "Video-based quantification of gait impairments in Parkinson's disease using skeleton-silhouette fusion convolution network," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 2912–2922, 2023.

[9] M. Lu et al., "Vision-based estimation of MDS-UPDRS gait scores for assessing Parkinson's disease motor severity," in *Proc. Med. Image Comput. Comput. Assist. Intervent.-MICCAI*, 2020, pp. 637–647.

[10] D. Wang, K. Yuan, C. Müller, F. Blanc, N. Padoy, and H. Seo, "Enhancing gait video analysis in neurodegenerative diseases by knowledge augmentation in vision language model," 2024, *arXiv:2403.13756*.

[11] M. Gholami et al., "Automatic labeling of Parkinson's disease gait videos with weak supervision," *Med. Image Anal.*, vol. 89, Oct. 2023, Art. no. 102871.

[12] M. Endo, K. L. Poston, E. V. Sullivan, L. Fei-Fei, K. M. Pohl, and E. Adeli, "GaitForeMer: Self-supervised pre-training of transformers via human motion forecasting for few-shot gait impairment severity estimation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, Jan. 2022, pp. 130–139.

[13] F. Yang, Y. Wu, S. Sakti, and S. Nakamura, "Make skeleton-based action recognition model smaller, faster and better," in *Proc. ACM Multimedia Asia*, Dec. 2019, pp. 1–6.

[14] M. Gholami, B. Wandt, H. Rhodin, R. Ward, and Z. J. Wang, "AdaptPose: Cross-dataset adaptation for 3D human pose estimation by learnable motion generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13065–13075.

[15] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks, "Deep generative modelling: A comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7327–7347, Nov. 2022.

[16] A. Martínez-González, M. Villamizar, and J.-M. Odobez, "Pose transformers (POTR): Human motion prediction with non-autoregressive transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 2276–2284.

[17] M. Petrovich, M. J. Black, and G. Varol, "Action-conditioned 3D human motion synthesis with transformer VAE," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 10985–10995.

[18] R. Dabral, M. H. Mughal, V. Golyanik, and C. Theobalt, "MoFusion: A framework for denoising-diffusion-based motion synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 9760–9770.

[19] C. Guo, X. Zuo, and S. Wang, "Action2Motion: Conditioned generation of 3D human motions," in *Proc. ACM Int. Conf. Multimedia*, 2020, pp. 2021–2029.

[20] L. Xu et al., "ActFormer: A GAN-based transformer towards general action-conditioned 3D human motion generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 2228–2238.

[21] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. NIPS*, vol. 33, 2020, pp. 6840–6851.

[22] M. Zhang et al., "MotionDiffuse: Text-driven human motion generation with diffusion model," 2022, *arXiv:2208.15001*.

[23] W. Feller. (1949). *On the Theory of Stochastic Processes, With Particular Reference to Applications*. [Online]. Available: https://api.semanticscholar.org/CorpusID:121027442

[24] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2256–2265.

[25] J. Ho and T. Salimans, "Classifier-free diffusion guidance," 2022, *arXiv:2207.12598*.

[26] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," 2020, *arXiv:2010.02502*.

[27] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 10684–10695.

[28] G. Tevet et al., "Human motion diffusion model," in *Proc. 11th Int. Conf. Learn. Represent. (ICLR)*, 2023. [Online]. Available: https://openreview.net/forum?id=SJ1kSyO2jwu

[29] Y. Bai, X. Wang, Y.-P. Cao, Y. Ge, C. Yuan, and Y. Shan, "DreamDiffusion: Generating high-quality images from brain EEG signals," 2023, *arXiv:2306.16934*.

[30] Y. Ji, Y. Yang, F. Shen, H. T. Shen, and W.-S. Zheng, "Arbitrary-view human action recognition: A varying-view RGB-D action dataset," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 289–300, Jan. 2021.

[31] Y. Ji, F. Xu, Y. Yang, F. Shen, H. T. Shen, and W.-S. Zheng, "A large-scale RGB-D database for arbitrary-view human action recognition," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 1510–1518.

[32] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1010–1019.

[33] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, Apr. 2018, pp. 7444–7452.

[34] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.