SEH-ConGAN: A Scalable GAN-based Framework for Robot-Assisted Automation in Animation Production

Hongping Tang
Animation and Game Academy, China Academy of Art, Hangzhou, 310024, Zhejiang, China
E-mail: tanghongping1989@hotmail.com

Keywords: robot-assisted animation, motion capture, automation, production pipeline, scalable elephant herding-tuned, conditional generative adversarial network (SEHConGAN)

Received: August 18, 2025

The animation production process is traditionally labor-intensive, requiring extensive manual effort in character motion design, scene composition, and post-production editing. To overcome these limitations, this research introduces a robot-assisted automation system integrated with artificial intelligence (AI) to streamline and accelerate animation development. The system incorporates a motion capture interface for acquiring human movement data, a feedback-enabled robotic arm to replicate and analyze motions, and a simulation environment for virtual testing. Preprocessing includes missing-value handling and Z-score normalization, after which structured motion sequences (3D joint coordinates, robotic servo positions, and torque data) are provided as input to the Scalable Elephant Herding-tuned Conditional Generative Adversarial Network (SEH-ConGAN). The model generates refined outputs such as smooth motion trajectories, facial expression synthesis, and context-aware style transfer. Statistical analysis using a paired t-test, 95% confidence intervals, and Cohen's d effect size was performed to confirm the significant performance improvement of SEH-ConGAN over baseline models Performance is evaluated using 5-fold cross-validation and achieves an accuracy of 0.96, precision of 0.97, recall of 0.96, and F1-score of 0.96. Comparative analysis of motion generation metrics shows that SEH-ConGAN surpasses existing models achieving the best MPJPE (16.7), FID (11.3), Smoothness (0.028), and Diversity (0.72), demonstrating superior motion accuracy, trajectory smoothness, and animation realism. . The findings demonstrate that combining robotics with SEH-ConGAN provides a scalable solution for producing high-quality animations with reduced time, cost, and manual intervention.

Povzetek: Raziskava predstavlja robotsko podprt AI-sistem za avtomatizacijo animacije, ki z modelom SEH-ConGAN omogoča natančnejše, bolj tekoče in realistične gibe ob bistveno manjšem času, stroških in ročnem delu.

1 Introduction

The animation production method is complex and multi-layered, with the production process converting creative ideas into action-rich visual content [1]. Pre-production covers storyboarding, concept design, and screenplay writing, and production includes character modeling, rigging, scene layout, animation, and rendering [2]. Lastly, compositing, sound design, and editing are the other post-production processes through which the output is made ready to be distributed. The technological advancements have augmented the productivity, but the traditional pipelines are still labor-intensive and require professional animators to take care of the subtle differences in the motions of the characters, and coordination of the scenes, and do some visual effects [3]. This makes animation both time-consuming and resource-intensive, especially for large-scale productions with high-quality standards.

Automation in animation refers to the utilization of electronic tools and computer procedures to make repetitive and time-consuming tasks easy [4]. It consists of procedural animation methods, automatic lip-synching, motion capture, and background making. Automation spares human effort in the tasks where manual repetition can be detrimental to production, and artists are able to focus on creative decision-making instead of technical performance [5]. With the introduction of artificial intelligence (AI) and machine learning, automation has been enhanced to include intelligent motion prediction, style transfer, and scene optimization, which have extended the production cycle and produced their products with guaranteed quality standards [6]. Due to the creative and subjective nature of animation, achieving full automation without human interaction is difficult.

Robot-aided systems are still automation beyond software and bring physical or creative support functions into the animation process that can be carried out by robotic hardware [7]. Such technologies could be applied to a stop-motion animation process to facilitate precise camera movements, robotically controlled puppets, and object placement [8]. The robotic installations within the virtual production setting could have the capability to collaborate with the digital technologies to capture the information regarding the motion, choose the movements of cameras, and reproduce the complicated shots and movements accurately [9]. Integration of this type enhances repeatability and reduces manual setup. Manual setup can be performed within, and fragile tasks can be repeated that are found by human operators to be quite impossible to perform manually. Figure 1 shows the robot-assisted animation production process.

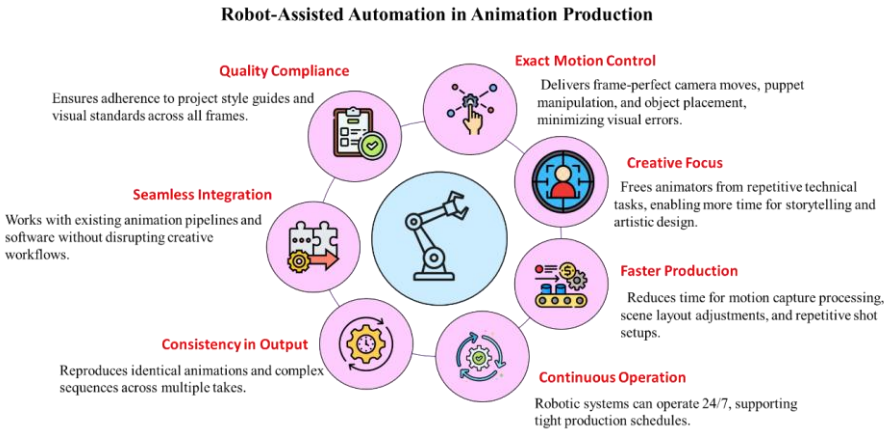

Figure 1: Robot-assisted animation production process

A robot-assisted automation system for animation production leverages modern robotics, AI, and animation software to maximize productivity while maintaining artistic integrity [10]. A configuration like this would be able to process every day, yet technically difficult tasks, make real-time adjustments, and promote the formation of a team between human artists and automatic tools by attaching sensors, motion control systems, and intelligent algorithms [11]. The strategy can transform the process of animation production through reducing costs, reducing schedules, and enabling studios to produce more creative material. Robot-assisted automation is one of the potential directions in a rapidly evolving entertainment industry to faster, smarter, and more adaptable animation pipelines [12].

Current machine learning techniques in animation production, including Long Short-Term Memory (LSTM) networks to predict temporal motion sequences and Variational Autoencoders (VAE) to generate smooth character poses, have increased automation but lack flexibility, compared to manual design, in that they are expensive to setup, can only adapt to particular artistic styles, and do not give as much creative freedom. These obstacles are overcome, to develop a SEH-ConGAN–based robot-assisted animation system that can learn human motion patterns, reduce motion estimation errors, and improve realistic motion style transfer to achieve higher adaptability, accuracy, and creative diversity in automated animation workflows. The contribution section is as follows:

Dataset: Prepared and validated good-quality animation motion dataset that comprised various motions of the character to adequately train the model.

Data preprocessing: It should be acknowledged that data preprocessing, Z-score normalization and missing values, was performed to ensure the stability of noise-free inputs into the model performance.

Proposed Framework: It suggested a robot-assisted animation system that effectively automates animation production phases by combining a deep learning model SEH-ConGAN.

Experimental validation: The proposed solution was tested experimentally and proved to be more animated, realistic, and performed better in terms of rendering than conventional manual production pipelines.

Research Questionnaire:

RQ1: In comparison to conventional ConGAN or manual animation techniques, can the SEH-ConGAN model produce robot-assisted animation sequences with better motion accuracy and stylistic fidelity?

RQ2: What effects do Elephant Herding Optimization (EHO)-optimized SEH-ConGAN hyperparameters have on training efficiency, style consistency, and animation quality?

RQ3: How much does the suggested approach enhance overall automation in the animation production process and lessen manual labor?

2 Related works

To examines developments in digital human technology and expression transfer in cinema and television animation [13]. Techniques like 3D digital human modeling and facial expression mapping improve emotional performance and realism. Results show increased viewer engagement and character plausibility. High computational costs, moral dilemmas, and possible abuse are among the limitations. There are documented numerical gains in motion precision and expression fidelity. It emphasizes the need of realistic, emotionally compelling animation in contemporary storytelling and calls for scalable, morally sound frameworks.

The deep learning (DL) technologies for animated scene creation and data mining were presented in [14]. It generated realistic and diversified animation scenarios using a powerful DL model, as well as Data Mining (DM) approaches such as cluster analysis and classification identification. The findings demonstrated DL's efficacy in lowering manual design burden and increasing efficiency. DM technology also offered appropriate market positioning and content innovation guidance for animation production.

Digital technology has revolutionized animation by incorporating physical human movement and bridging the gap between digital and performing arts. A framework of real-time character animation that integrated the performer as an instantaneous creator of effect, expression, and character was proposed in [15]. It was possible to visualize the internal response through wearable technology by capturing bio signals (i.e., heart

rate and skin response). Movement and bio signals were captured by sensors to create nonverbal personality characteristics and signs.

The framework allowed humanoid robots to develop expressive motion sequences called EMOTION [16], which improved the capacity to participate in nonverbal communication. The method employed huge language models' in-context learning to build socially suitable gesture motion sequences for human-robot interaction. The system was evaluated and shown to meet or outperform human performance in creating intelligible and natural robot motions.

The animated film characters improved the quality and accuracy of the pictures in the first-order motion model (FOMM). Convolutional block attention model (CBAM) was proposed in [17], who wanted to think about the essential features and restore the distortion of the image due to the change of posture. The suggested enhanced FOOM (E-FOOM) model was intended to improve end-to-end character image production. According to the experimental results, the E-FOOM model had the highest performance of image resolution, key point detection accuracy, and reconstruction of posture when compared to other models.

AI and machine learning have revolutionized animation, altering character movement and interaction [18]. It investigated how the technologies automate chores, increase realism, and open up new creative possibilities. It examined scenarios and industry practices to demonstrate

The influence of AI on storytelling, production pipelines, and the prospects of animated entertainment. It recognized the limitations of AI in animation, which could influence animators' careers and enterprises. Table 1 displays the further related works.

Table 1: Comparative summary of existing animation and robot-assisted motion generation techniques

References	Technique / Model Used	Dataset / Input Type	Evaluation Metrics / Results	Strengths	Limitations
Racinskis et al. 2022 [19]	Feedforward NN & Recurrent Neural Networks for robot motion concepts	Multi-modal inputs; Motion capture (most feasible)	RNN outperformed FFNN but not consistently	Demonstration-based robot motion learning; multi-modal capability	Performance not stable; lacks style transfer; no robotic animation generation
Liu et al. 2024 [20]	CAD + Reinforcement Learning + Computer Vision for animation SFX	CAD models, RL action modeling, CV-based object tracking	Generated realistic SFX during filming	Integrates CAD, RL, CV for automated SFX creation	Focus only on SFX; does not handle motion style transfer; no robot-assisted system
Wang et al. [21]	Deep Reinforcement Learning with RAG + dynamic reward	Rule-based action inputs for swarm animation	Improved swarm behavior & control precision	High-quality swarm animation; real-time interaction	Not suitable for human motion prediction; no GAN-based realism enhancement
Pibernik et al. 2023 [22]	Experimental evaluation of loading-animation design, semantics, and motion properties	Loading animations with varied structure, metaphor, and speed	Statistical tests showed significant influence on perceived wait-time; measurable differences in perceived load-speed	Demonstrates importance of nontemporal animation cues; strong empirical insights	Conducted in controlled laboratory setting; limited animation variations
Kang and Kim et al. 2024 [23]	CAD-based character modeling + VR-driven action design + optimization algorithm	CAD geometric models, VR-based interaction sequences	High-quality character outputs; real-time performance; high user satisfaction across simulations	Integrates CAD, VR, and optimization; realistic action generation; strong real-time capability	Limited simulation diversity; no statistical validation across diverse animation environments

Li et al. 2025 [24]	Temporal-Stylistic Latent Animator (TSLA) + Domain-Informed Animation Realignment Strategy (DIARS)	Real-time animation sequences capturing temporal and stylistic dynamics	Outperforms existing methods in animation completion, style transfer, and semantic editing	High-fidelity synthesis; semantic consistency; stylistic coherence; supports real-time, complex human motion	Computationally intensive; requires comprehensive training datasets; may need domain-specific tuning

2.1 Problem statement

The existing animation approaches have significant limitations. Motion capture technologies [13] often face high prices, complicated integration into production workflows, and demand substantial resources, restricting accessibility. Deep learning-based animated scene production [14] can save manual design labor, but it cannot be adaptable to different animation styles and requires a large amount of labeled data. Reinforcement learning techniques for animation special effects [20] enhance automation and quality, but they frequently require significant computer resources and suffer from real-time performance restrictions. To address these issues, the proposed SEH-ConGAN provides a scalable and efficient framework that uses elephant herding optimization to fine-tune the conditional GAN, improving animation quality while lowering training time and resource requirements. This technology strikes a compromise between accessibility and high-quality output, hence improving practical animation production procedures.

3 Methodology

The process of animation production has never ceased to be labor intensive; this includes the proper motion design, structure of the scene and the post production. It uses a curated dataset of animation motion that contains a great variety of motions, which were first preprocessed with the aid of missing value management and Z-score normalization to give uniform noise-free data. The presented proposal is a robot-assisted animation system, which automated and optimized production stages involving the integration of the SEH-ConGAN. Experimental results depict high promotion of evaluation measures, which is a scalable cost-effective option of studios and individual producers seeking high quality productions without manual interventions. Figure 2 shows the overall suggested flow for the animation production process.

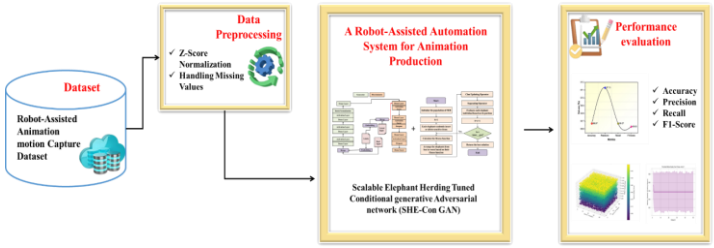

Figure 2: Overall suggested flow for robot-assisted animation production process

3.1 Dataset

The Robot-Assisted Animation Motion Capture Dataset combines motion capture, robotic arm input, and motion quality evaluation to provide detailed, realistic data for AI-driven animation development. It is composed of 100 motion sequences, each comprising 50 frames of 3D skeleton joint coordinates, servo motor locations, and torque data from a simulated robotic arm. The dataset encompasses a range of motion types, including walking, running, leaping, waving, sitting, and dancing, along with additional information such as actor ID and a motion_quality_score column for supervised learning or regression purposes. Its multi-modal and well-structured data format makes it perfect for investigating motion tracking, robotic animation control, movement analysis, and animation quality evaluation, allowing for repeatable trials with controlled variability.
Source: https://www.kaggle.com/datasets/ziya07/robot-assisted-animation-motion-capture-dataset/data

3.2 Data preprocessing

Data preprocessing in robot-assisted animation production includes imputation or removal of missing values to ensure dataset completeness, as well as Z-score normalization to standardize features and enable consistent scaling for improved AI model accuracy and performance during motion tracking and scene automation.

3.2.1 Handling missing values

To handle missing values in the robot-assisted animation motion capture dataset, first check for gaps in the 3D joint coordinates, servo motor locations, torque measurements, and metadata fields. Interpolation or forward/backward filling can be used to approximate missing frames in sequential numerical data such as joint coordinates and sensor inputs while maintaining temporal continuity. If the missing data is large or impacts whole sequences, consider deleting those samples to preserve dataset

quality. If categorical metadata, such as motion type or actor ID, is missing, check the source logs or eliminate the impacted sequences. This method minimizes data loss while retaining integrity, resulting in reliable motion analysis and machine learning tasks.

3.2.2 Z-Score normalization

In the robot-assisted animation motion capture data set, the focus will be on each numerical feature to its mean (W) and normalized to its standard deviation (σ). This will be done according to the following formula: The data are normalized to maintain consistent scaling across the features (including 3D coordinates of skeleton joints, servo motor position, and torques) in all motion sequences, regardless of their initial point units or value range. The approach increases model stability, learning efficiency, and detects tiny variations in motion quality patterns, allowing for precise robotic animation control and quality evaluation.

$$W_{new} = \frac{W-\mu}{\sigma} \quad (1)$$

Where W_{new}- new value, W – old value, μ- mean, σ- standard deviation value. The servo locations and torques following Z-score normalization are shown in Figure 3, which ensures consistent feature scaling, balanced value ranges, and increased comparability for accurate analysis in robot-assisted animation production.

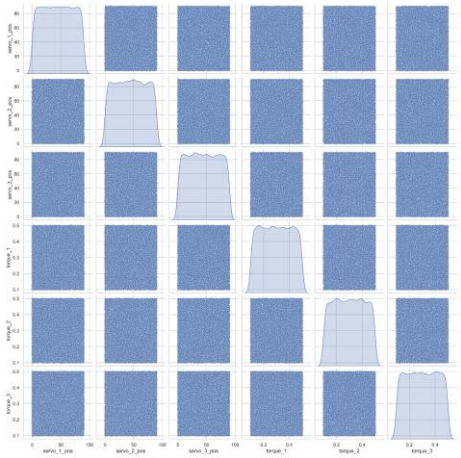

Figure 3: Pairwise feature relationships after Z-score normalization for servo positions and torques

3.3 SEH-ConGAN

The SEH-ConGAN is a novel approach that revolutionizes the field of robot-aided animation production, combining AI in content production with smart optimization. Traditional animation production involves a heavy investment of manual work in designing character movements, setting up the scenery, and characterizing the scene dimensions, and this might be long and tedious. SEH-ConGAN addresses these limitations effectively to enable the robotic arm executes motion patterns for animation generation, which can adjust styles to various plots and dynamically respond to director instructions through changes in motion patterns. By using conditional GANs, the system creates outputs based on certain animation parameters (e.g., position, camera angle, illumination), resulting in context-aware creation. The Elephant Herding Optimization (EHO) method fine-tunes hyperparameters for optimal training stability, frame quality, and style consistency, allowing the system to scale across animation genres and production sizes. Due of their slowness and frequent instability for GANs, SEH is preferred over grid search and Bayesian optimization. SEH provides smoother motion with better animation quality, increases convergence speed, and more effectively investigates hyperparameters. Elephant Herding Optimization (EHO) in SEH-ConGAN automatically modifies the GAN hyperparameters, such as learning rate, network size, and noise settings, to improve training. Selecting these hyperparameters by hand may be sluggish, erratic, or result in frames of poor quality. EHO functions similarly to an elephant herd, with leaders directing sound solutions and substituting poor alternatives. Compared to standard GAN tweaking, this lets the model explore hyperparameters more effectively, trains more quickly, lowers frame mistakes, enhances style consistency, and produces more accurate, high-quality animations. The Discriminator C tries to maximize U(C,H) making real frames score high and fake frames score low. SEH-ConGAN uses the Generator (H) to produce real animation frames as w based on conditioning variables z, and the Discriminator (C) to discriminate between actual and created frames (equation 2).

$$\min \max U(C,H) = F_{w,z}\ [\log C(w|z)\ ]|\ F_{w,z}\ [\log\ \ (1\ \ C(H(w|z)|z))\ ]\ (2)$$

EHO optimizes learning rates, architectural depth, and noise parameters to reduce frame reconstruction error in equation (3).

$$\text{Fitness}=\frac{1}{M}\sum_{j=1}^{M}\left\|\hat{w}_j-w_j\right\|^2 \quad (3)$$

Where M is the number of frames, $\hat{w}_j$ is the generated frame, w_j is the real frame, and $\left\|\hat{w}_j-w_j\right\|^2$ measures the pixel-wise error to evaluate fitness. Algorithm 1 shows the pseudocode for SEH-ConGAN.

Advantages of using SEH-ConGAN

- Reduces animation production time by automating repetitive creative tasks.
- Creates frames that are both style-consistent and contextually aware.
- Scalability allows for the adaptation to numerous animation genres.
- Improves training efficiency with clever hyperparameter adjustment.
- Improves robotic coordination for complicated scenario execution.

Algorithm 1: SEH-ConGAN

Input:
- *Dataset D with 10,000 samples and 10 classes*
- *Number of clans C = 3*
- *Clan size N = 5 elephants per clan (total 15 elephants)*
- *Maximum iterations MaxIter = 20*
- *GAN training epochs per evaluation = 5 epochs*
- *Hyperparameter search space P:*
  - *Learning rate: [0.0001, 0.001, 0.01]*
  - *Batch size: [32, 64, 128]*
  - *Noise dimension: [50, 100, 150]*

Output:
- *Trained conditional GAN with optimized hyperparameters*

Begin
1. Initialize elephant population E:
  For each clan c in [1..3]:
    For each elephant j in [1..5]:
      Randomly assign hyperparameters h_e from P
      e.g. h_e = {learning_rate=0.001, batch_size=64, noise_dim=100}
2. For iteration = 1 to 20 do:
  2.1 For each elephant e in population E:
    - *Extract hyperparameters h_e = {lr, batch_size, noise_dim}*
    - *Train a conditional GAN for 5 epochs on dataset D using h_e*
    - *Evaluate performance metric f_e = FID score on validation data (Lower FID means better)*
  2.2 For each clan c in [1..3]:
    - *Find clan chief e_chief with lowest FID in clan c*
    - *For each clan member e_j ≠ e_chief:*
      Update h_e_j as:
        lr_new = lr_chief + random_uniform(-0.0001, 0.0001)
        batch_size_new = batch_size_chief (round if needed)
        noise_dim_new = noise_dim_chief + random_choice([-10, 0, +10])
      Ensure h_e_j within valid range
  2.3 Clan separation:
    - *For each clan:*
      Remove worst performing elephant (highest FID)
      Replace with a new elephant with random hyperparameters from P
3. After 20 iterations:
  - *Select best elephant e_best with lowest FID*
  - *Retrain the conditional GAN on full training data for 50 epochs using e_best hyperparameters*
4. Return trained GAN model and e_best hyperparameters
End

3.3.1 ConGAN

The ConGAN modules, with a special emphasis on the two primary modules, generator and discriminator, and their responsibilities in the animation creation process. In this case, the generator acts as a creative design team, creating frames, characters, and scenarios based on the provided concept art and style guides. The discriminator serves as a quality control unit, examining each produced sequence to verify that it satisfies the intended creative style, smoothness of motion, and visual consistency before it is incorporated into the final animation. In Figure 4, the ConGAN model takes as input a latent noise vector

and motion type labels, along with real motion capture sequences from the Robot-Assisted Animation Motion Capture Dataset. It outputs either generated motion sequences (Generator) or a probability score (Discriminator) indicating whether the input motion is real or AI-generated, enabling realistic animation synthesis and validation.

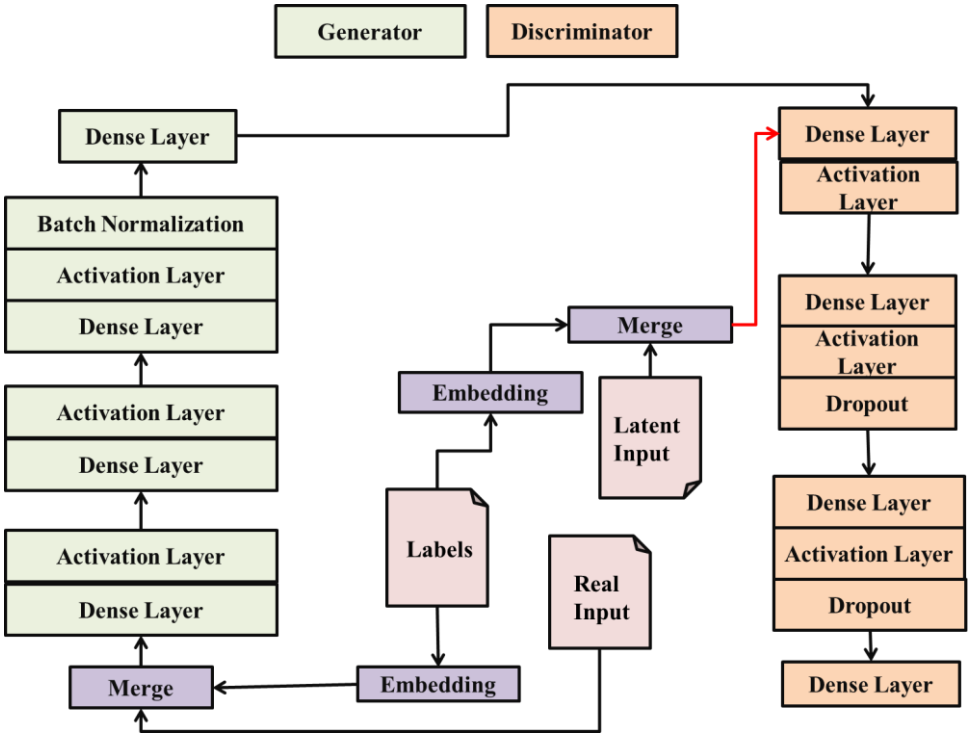

Figure 4: ConGAN architecture

Generator

The generator is a stack of artificial neural networks where the number of cells doubles in each successive layer. Activation layers are separated by dense layers. The last layer is that which comes preceding the output layer, which is a batch normalization layer that is used as a regularization measure to enhance stability and generalization properties. Equation (4) supplies a generator function to the latent input y and a conditional input d, where H is a nonlinear function, such as an artificial neural network. Equation (4) produces a sequence of multivariate data called $\vec{z}_h$ that represents the animation material being modeled throughout the creation process.

$$H:(\vec{y},\vec{d})\mapsto\vec{z}_h \quad (4)$$

The vector $\vec{z}_h$ could represent discrete, continuous, or a combination of both type of variable. To prevent mode collapse and overfitting, white noise is added to the $\vec{z}_h$ output before feeding it to the discriminator.

Discriminator

It has been shown that other assumptions include the fact that the log-likelihood-ratio between created animation frames $\vec{z}_h$, and the actual frames $\vec{w}_c$ is finite, and the Jensen-Shannon divergence will not reach a maximum value under the condition $JS[\vec{z}_h|\vec{w}_c]$. This leads to the use of additive noise with a normal distribution and variable variance to avoid over-fitting of the quality evaluator (discriminator) during training. It introduced noise to each sample, $z_h \in \vec{z}_h$ and $w_c \in \vec{z}_c$, to avoid overfitting on the training set (equation 5).

$$C:(\vec{z},\tilde{w})\mapsto\vec{z}_c \quad \text{where} \quad \vec{z}\supseteq\vec{z}_h,\vec{w}_c \quad (5)$$

As part of the training, the objective loss function of the ConGAN, which measures the connection between the generator and discriminator in the generation of animation, is given as follows in equation (6), where K_H and K_C are losses of the generator and discriminator.

$$K_C=F\left[\log\left(C(\vec{w}_c,\tilde{w})\right)\right]+F\left[\log\left(1-C\left(F\left(H(\vec{y},\vec{d}),\tilde{w}\right)\right)\right)\right]$$
$$K_H=F\left[\log\left(C\left(H(\vec{y},\vec{d}),\tilde{w}\right)\right)\right] \quad (6)$$

Where C is the discriminator, H is the generator, $\vec{w}_c$ is the real input data, $\vec{y}$ is the encoded noise vector, and $\vec{d}$ is the encoded condition or context. K_C evaluates the discriminator's ability to discriminate between actual and created samples, whereas K_H measures the generator's ability to mislead the discriminator.

Controlling discriminator overfitting

It is possible to add synthetic noise to the output of the generator ($\vec{z}_h$) and the real sample input ($\vec{w}_c$) and use this in animation production. In this case, the generator could represent a robot-based animation frame generator, whereas the real scenarios are legitimate, artist-created frames. One of the basic assumptions of GANs is that

the logarithmic probability ratio $\frac{\vec{z}_h(\vec{z}_c)}{\vec{w}_c(\vec{z}_c)}$. However, in complex animation production scenarios, given $H:(\vec{y},\vec{d})\mapsto\vec{z}_h$ where the support is $\{y\in\vec{y},d\in\vec{d}:H(y,d)\neq 0\}$, the intersection between the generator's output space and the real frame distribution $\vec{w}_c$ in high-dimensional feature space could be $\emptyset$ if the distributions are degenerate. As shown in equation (5), the addition of synthetic noise would generate overlapping supports of these two distributions. This overlap ensures that the probability distribution is not infinite, and the Jensen-Shannon divergence is a continuous function and will not lead to a final result that is constant. As a consequence, overfitting in the discriminator is minimized, resulting in more consistent and visually cohesive animation frames.

3.3.2 Scalable elephant herding (SEH)

SEH in animation production makes it easier to manage significant, complicated character groups by allowing for effective coordination, realistic movement modeling, and resource optimization for crowd scenes, resulting in faster production times and better visual consistency in large-scale animated productions. Elephants are gregarious creatures that live in tribes and are female. A matriarch leads an elephant clan, and it includes a large number of elephants. The female elephants in the clan decided to stay with their family members, whilst the male elephants prefer to be outside.

They gradually grow independent of their family, eventually leaving them. SEH is inspired by elephants' herding habits. SEH considers the following assumptions:

- Some clans have a predetermined number of elephants.
- Male elephants were able to leave behind their group of family and live alone for generations without being integrated into the rest of the group.
- Every clan has a matriarch who leads the elephants.

Elephant behavior could be modeled as clan updates and separations. Each elephant in the population adds or removes creative elements in production tasks to ensure animation quality and consistency. SEH models elephant behavior as clan updating and separating operators. The clan-modifying operator updates the elephants' present location and matriarch, followed by the separating operator. The suggested SEH model aims to optimize production speed while balancing refinement (exploitation) and innovation (exploration) stages.

The initialization process and fitness function

A clan contains several elephants, and each elephant is a solution (i.e., a completed animation sequence), again represented using a series of 0s and 1s. The 1 denotes an appearance of a critical animation frame, whereas the 0 denotes its absence from the sequence. The first elephant in the population symbolizes a series of key frames from the original storyboard. The remaining elephants in the population update specific frames at random, but the initial elephant's frames stay unchanged. As a result, an initial population including a variety of animation options is created. Following the initialization procedure, each elephant's fitness value is computed based on the smoothness of motion, consistency with the storyline, visual coherence, scheduling accuracy, and frame continuity. The fitness function is expressed in equation (7).

$$\text{Min } \overrightarrow{fit}=[fit_1,fit_2,fit_3,fit_4,fit_5] \quad (7)$$

Where,

$$fit_1=|HF|$$
$$fit_2=|LR|$$
$$fit_3=RHD+RLD$$
$$fit_4=\frac{N0\_of\_GR}{R}$$
$$fit_5=\frac{No\_of\_T}{Size\_of\_C}$$

In the equation above, |HF| indicates the number of hidden flaws, |LR| indicates the number of lost revisions, the revision hiding distance is RHD, and the revision loss distance is RLD. No_of_GR denotes the number of ghost frames, which are animation frames that were not included in the original storyboard but emerge in the final render following revisions. R is the total number of frames completed that meet the required Minimum Quality Threshold (MQT) and Minimum Consistency Threshold (MCT). No_of_T indicates the number of adapted scenes, and Size_of_C represents the whole size of the animation production.

The clan upgrading operator

Each elephant z in clan w has an old place ($f^s_{w,z}$). The new position will be affected by the matriarch of the clan, $f^{s+1}_{w,z}$, n^s_w Matriarch's position for the clan w at iteration s, using the following equation in equation (8):

$$f^{s+1}_{w,z}=f^s_{w,z}+\alpha\times\left(n^s_w-f^s_{w,z}\right)+\beta\times\left(d^s_w-f^s_{w,z}\right)+\gamma\times rand \quad (8)$$

Where α, β, and γ are scaling variables ranging from 0 to 1 that define the task effect on an animator's new position, the animator's affinity towards the core production team, and the animator's inclination to function autonomously, respectively. The random vector rand = $(2\times r-1)(f_{max}-f_{min})$ is chosen from a uniform distribution, with f_{max} and f_{min} representing an animator's upper and lower constraints on the animation parameters. d^s_w is the center of the production team and is determined as follows in equation (9):

$$d^s_w=\frac{1}{Num_w}\times\sum_z f^s_{w,z} \quad (9)$$

Where Num_w is the number of elephants in the Clan w. The new position of the matriarch is a straight mix of their previous position. To control convergence on the clan center, as well as that of the random walk, three control parameters (alpha, beta, and gamma) are used.

Separating operator

Male elephants create a separation operator, which could be simulated.

$$f^s_{w,worst}=f_{min}+(f_{max}-f_{min})\times q \quad (10)$$

Equation (10) translates f_{min} and f_{max} as the upper and lower limits of an elephant's position, respectively, and $f^s_{w,worst}$ as the worst single elephant in the clan d_w. The

separation operator probability density function starts with q, a pseudo-random number Generator (PRNG), a uniformly distributed random number between 0 and 1 generator. To produce a uniformly distributed, arbitrarily-chosen integer between the boundaries [f_{min} , f_{max}], q must be scaled and moved. The floor function is utilized to generate an evenly distributed arbitrary integer value within a given range. The floor function $\left([f_{min}, f_{max}) \right) = (f_{min}. f_{max-1})$ indicates a constant homogeneous distribution over the range [f_{min}, f_{max}]. Figure 5 depicts the whole flow of SEH-based optimization for scheduling the animation production process.

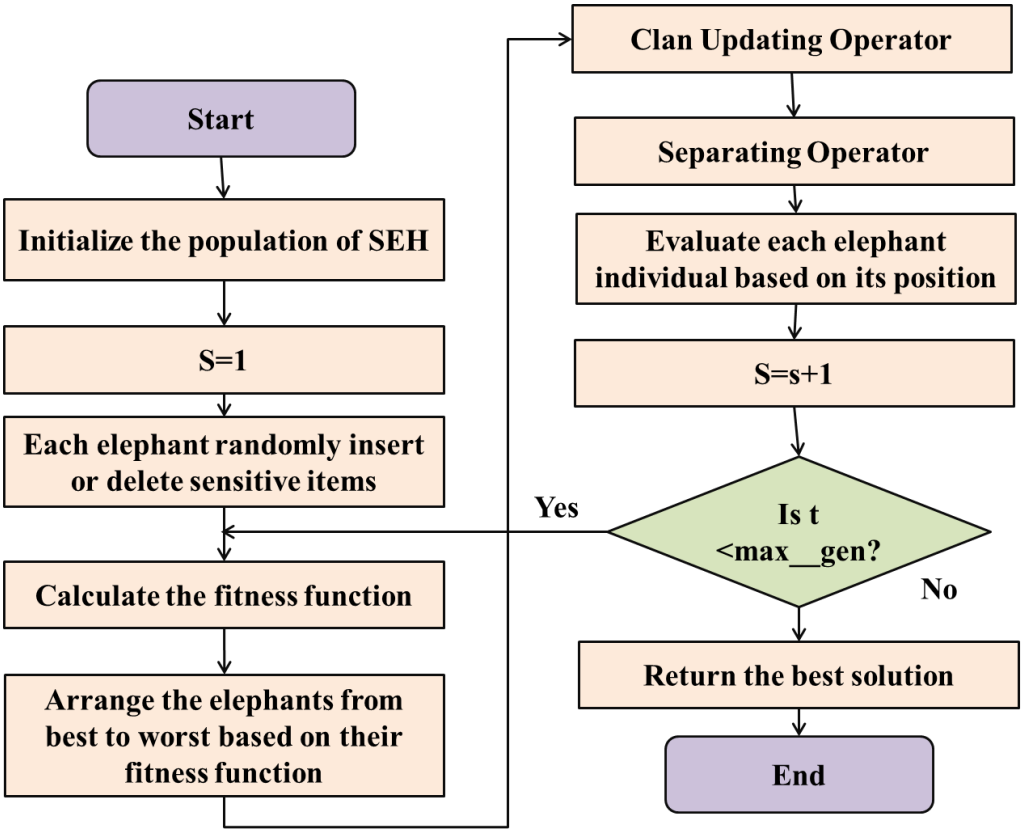

Figure 5: Flowchart for SEH

The hyperparameters of the Scalable Elephant Herding-tuned Conditional GAN (SEH-ConGAN) framework are described in Table 2. It covers parameter symbols, example values, ranges, and explanations for both GAN training and SEH optimization, including learning rates, population structure, update frequencies, and fitness assessment parameters for balanced generator-discriminator performance
.

Table 2: Hyperparameters for SEH-ConGAN

Hyperparameter	Symbol / Vector Pos.	Example Value	Range / Bounds
Total epochs	E	200	-
Batch size	B	64	16 - 128
Discriminator steps/iteration	n_D	1	1 - 5
Generator steps/iteration	n_G	1	1 - 5
SEH period (epochs)	P_seh	10	5 - 20
SEH population size	N_pop	24	12 - 40
Number of clans	C	4	3 - 6
Clan size	-	6	-
SEH iterations per run	I_seh	12	5 - 30
Candidate dimension	dim_p	6	-
Generator learning rate	p[0] = lr_G	0.0002	1e-5 - 5e-3
Generator β_1	p[1] = beta1_G	0.5	0.0 - 0.999
Latent std. scale	p[2]	1.0	0.5 - 2.0
BatchNorm momentum	p[3]	0.1	0.01 - 0.5

Conv. channel scale factor	p[4]	1.0	0.5 - 2.0
L2 reg. lambda	p[5]	0.0002	0.0 - 1e-3
Clan attraction factor	α	0.12	0.05 - 0.2
Gaussian perturbation scale	β	0.02	0.01 - 0.05
Separation probability	p_separate	0.12	0.05 - 0.2
Migration frequency	migration_freq	3	2 - 5
Fitness class accuracy weight	γ	0.8	0.5 - 1.0
Inner steps in candidate eval	inner_steps	3	1 – 8

4 Results and discussion

The implementation details display in Table 3 ensure reproducibility and provide a clear understanding of the computational resources required for the proposed framework.

Table 3: Implementation and training specifications

Component	Description
Hardware (Computational Setup)	NVIDIA RTX 3080 GPU (10 GB), Intel i7-11700 CPU, 32 GB RAM, Ubuntu 20.04
Framework	Python, PyTorch 2.0
Training Time per Epoch	24 seconds (SEH-ConGAN), 19 seconds (ConGAN baseline)
Total Training Duration	~78 minutes for 200 epochs
Stopping / Convergence Criteria	Stop after 200 epochs OR generator loss $\Delta < 0.001$ for 12 consecutive epochs
Generator Architecture	7 layers, 3.2M parameters
Discriminator Architecture	5 layers, 1.1M parameters
Total Parameters	4.3M trainable parameters
GAN Training Strategy	1 discriminator step per iteration, 1 generator step per iteration

To assess the proposed technique, with the standard ConGAN trained on the robot-assisted animation motion capture dataset serving as a baseline comparison. The dataset contains several motion patterns from robot-assisted animation creation. It enables quick preprocessing, training, and evaluation, ensuring fair performance evaluation for creating high-quality, realistic animated motion sequences.

4.1 Experimental results

The robot-assisted animation production process was depicted in Figure 6. Figure 6(a) depicts 3D motion trajectories for activities such as running, sitting, waving, jumping, walking, and dancing in X, Y, and Z space, allowing the system to capture, distinguish, and accurately replicate diverse character movements during animation creation. It serves as a visual reference for motion variety, allowing users to spot motion overlaps and distinct movement patterns. This information enables the robot to automatically adjust animation environments to match Individual motions, resulting in realistic and expressive output. Figure 6(b) shows the correlation between the position of servo motors (Servos 1 and 2) and the rating of the motion quality, in which the brighter colors represent higher quality. It draws the optimal control range over the servo motions that produce the most fluid animations. This kind of calibration guarantees reduced mechanical stress, longer component life and stability in the reproduction of high-quality motion.

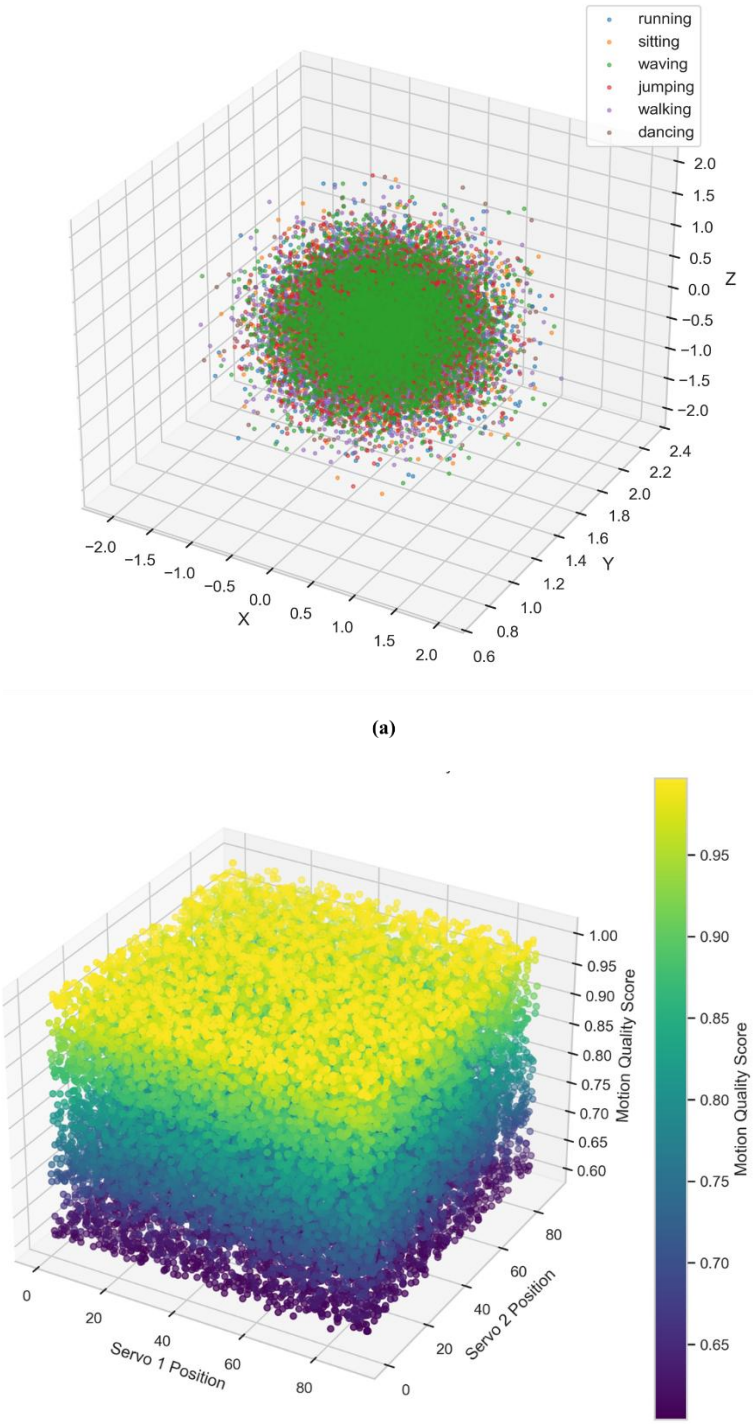

Figure 6: Robot-assisted animation production process (a) motion trajectories for activities (b) servo position vs motion quality

Figure 7 represents the torque patterns (torque_1, torque_2, and torque_3) of three actuators during 50 animation frames in the process of creating the animation involving the robot-assisted. The values of torque remain at the highest limit (~0.5) and change slightly but in the same way, which also demonstrates the stable and steady functioning of the joints. It is stable and thus the motion can be executed smoothly and accurately and this is very important in creation of good animations. The observation and analysis of these torque patterns can allow a more balanced distribution of loads, reduced mechanical wear and better performance of the actuators. It makes robots work reliably and efficiently throughout long sequences of animation production through the refinement of control algorithms.

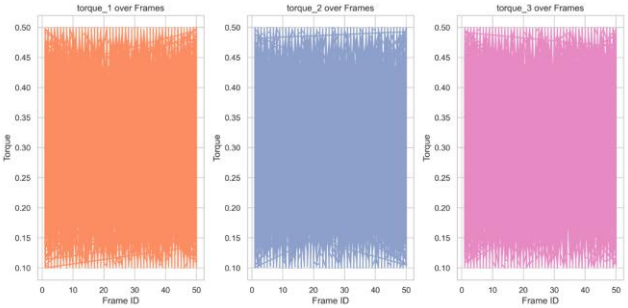

Figure 7: Torque variation across frames for robotic actuators in animation production

The column actor-id is probably the one that is the person (human or robot) who carries out the motion sequence recorded. The Figure 8 shows the performance measures of Actor 1 during the process of creating robot-assisted animation. The Z-axis motion versus time is plotted in figure 8(a) where the red line illustrates changes in positions with 50 frames and the shaded area represents variation. It assists in calculating the stability of horizontal motion leading to smooth and natural motion. Figure 8(b) shows the smoothed quality of motion frame by frame with a rolling mean, the purple line shows a constant quality and the stippled area shows moderate changes. With these insights, purposeful adjustments to the parameter of robotic motion are possible, minimizing deviations of the expected movements. Such analysis guarantees accurate motion replication, which improves animation realism and production efficiency.

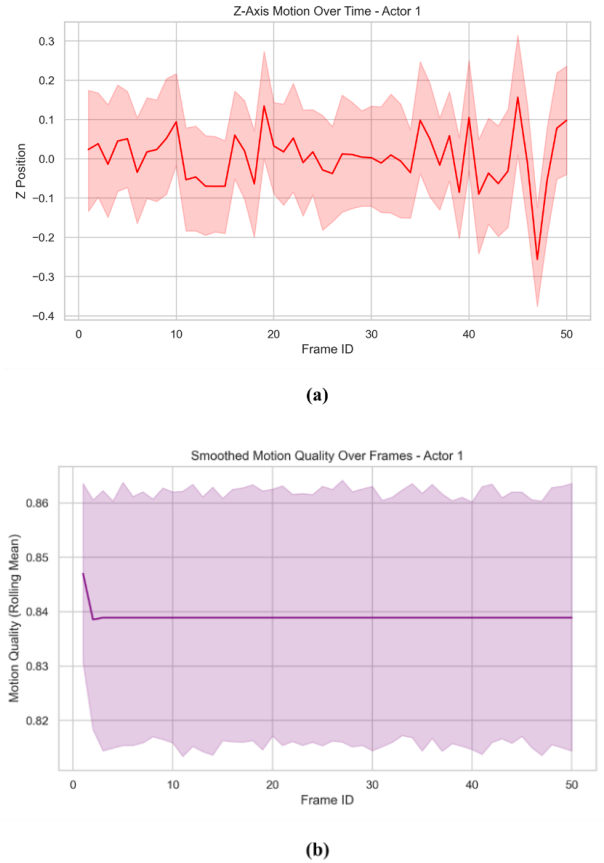

Figure 8: Actor 1 performance for animation production process (a) Z-axis motion over time (b) smoothed motion quality over frame

4.2 Performance metrics

The performance metrics for robot-assisted animation creation are interpreted using frame-level and action-level validation:

- Accuracy: The percentage of created animation outputs that accurately match the ground-truth motion plan, including anticipated poses, transitions, or scene-level activities, is known as

accuracy. It shows how reliably the technology generates legitimate motion sequences.

- Precision: The percentage of correctly generated frames or actions among all outputs that the model classifies as accurate is known as precision. It assesses how well the system ensures that the majority of created motions are legitimate by preventing the production of erroneous, jittery, or stylistically inconsistent frames.
- Recall: It quantifies the number of crucial motion transitions, animation frames, and important ground truth acts that are successfully replicated. A high recall means that crucial stages or movement components are not overlooked by the system.
- F1 Score: The model's overall capacity to generate error-free frames while simultaneously capturing all necessary motion features is shown by the F1-score, which strikes a compromise between precision and recall.

The translation of classification metrics to animation creation is made evident by treating each created frame or motion action as an anticipated output and comparing it with the ground-truth animation sequence.

The stance of each frame is compared to the ground-truth pose in SEH-ConGAN since the model predicts animation at the frame level. When the difference in joint angles is between two and five percent of the actual motion, a frame is considered accurate. In order to preserve smooth motion, we also verify that action transitions (such as steps or rotations) occur in the proper sequence. Since the model produces pose sequences rather than images, pixel-level comparisons are not utilized. With this configuration, the number of correct frames, the number of incorrect frames avoided, and the completeness of the created motion sequence are all directly measured by accuracy, precision, recall, and F1-score.

4.3 5-Fold cross-validation for animation production process

The 5-Fold cross-validation for the robot-assisted animation production process reveals remarkable performance in automating motion design, scene coordination, and post-production operations using SEH-ConGAN, as shown in Table 4. Across five folds, the system obtained perfect results in folds 1 and 4, with accuracy, precision, recall, and F1-score all equal to 1.0000, signifying perfect execution. The scores for fold 2 were accuracy 0.9667, precision 0.9722, recall 0.9667, and F1-score 0.9664, whereas fold 3 obtained 0.9333, 0.9400, 0.9333, and 0.9325. Fold 5 measured 0.9333, 0.9444, 0.9333, and 0.9324 for similar parameters. These consistently good results show the AI-driven system's capacity to reliably forecast motion patterns, manage scene variances, and provide smooth, high-quality outputs, minimizing the need for manual intervention and simplifying processes in animation creation.

Table 4: Performance metrics using 5-Fold cross-validation and their average values for SEH-ConGAN

Fold	Accuracy	Precision	Recall	F1-score
1	1.0000	1.0000	1.0000	1.0000
2	0.9667	0.9722	0.9667	0.9664
3	0.9333	0.9400	0.9333	0.9325
4	1.0000	1.0000	1.0000	1.0000
5	0.9333	0.9444	0.9333	0.9324
Average values	0.96	0.97	0.96	0.96

The robot-assisted automation system for the animation production process performs well using SHE-ConGAN, with an accuracy of 0.96, showing that it successfully automates motion design and scene coordination in almost all circumstances. The model's precision of 0.97 demonstrates its ability to produce correct and artistically consistent animations with minimal faults. A recall rate of 0.96 demonstrates its ability to capture and complete a significant number of essential animation tasks without exclusion. The F1-score of 0.96 demonstrates a fair trade-off between precision and recall, resulting in dependable, efficient, and high-quality animation automation.

4.4 Comparison of the proposed technique with standard techniques

Figure 9 illustrates that the proposed SEH-ConGAN method is better than its counterpart, the ConGAN in the robot-aided animation generation procedure. In Figure 9, SEH-ConGAN enhances accuracy of motion of robots by minimizing position error by 4.5cm and4.9o to 1.8cm and 2.3o respectively. It also enhances the smoothness of trajectory by 78 to 95, motion repeatability by 72 to 96 and pose alignment score by 74 to 94. The performance of style transfer is indicated in Table 5, where SEH-ConGAN led to higher animation style fidelity of 94% compared to 81%, color consistency of 92 over 79, and texture/detailed accuracy compared to 91 over 77, poses style consistency compared to 93 over 80 and visual consistency across frames compared to 78 over 72, indicating better animation clarity and consistency. The aforementioned criteria measure how realistic robot-assisted animation is: visual coherence guarantees seamless temporal transitions, while style integrity, color, texture, and posture accuracy capture spatial correctness. SEH-ConGAN consistently outperforms ConGAN in every metric, exhibiting improved style transfer and visual authenticity. These findings show that SEH-ConGAN not only improves robotic motion execution but also provides high-quality visual style transfer. Overall, the strategy considerably improves mechanical correctness and visual authenticity in robot-assisted animation production.

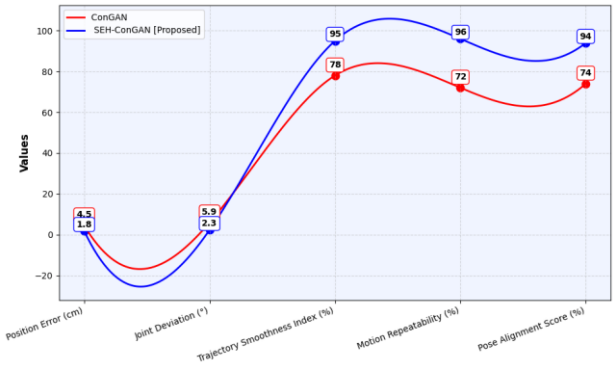

Figure 9: Comparative performance for the proposed and the baseline method using the animation production process, robotic motion accuracy

Table 5: Performance analysis of style transfer accuracy baseline and proposed method

Style Transfer Accuracy	ConGAN	SEH-ConGAN [Proposed]
Animation Style Fidelity (%)	81	94
Color Consistency (%)	79	92
Texture & Detail Accuracy (%)	77	91
Pose Style Matching (%)	80	93
Visual Coherence Across Frames (%)	78	92

4.5 Performance comparison of existing and proposed motion generation models

The performance of existing motion generation models Time-Series Latent Adversary (TSLA), Action-Conditioned Transformer for Motion Generation (ACTOR), Motion Generative Flow (MoGlow), and Video Swin Transformer – generative variant (VideoSwin) [24] was compared with the proposed Scalable Elephant Herding-tuned Conditional Generative Adversarial Network (SEH-ConGAN) for robot-assisted animation. Four key evaluation metrics were used: Mean Per Joint Position Error (MPJPE), Fréchet Inception Distance (FID), Smoothness, and Diversity as shown in Table 6.

MPJPE (Mean per Joint Position Error)

MPJPE evaluates motion correctness in animation by calculating the average Euclidean distance between expected and ground-truth 3D joint locations. More accurate motion replication is indicated by lower numbers in Equation (11).

$$MPJPE=\frac{1}{N}\sum_{i=1}^{N}\left\|P_j^{pred}-P_i^{gt}\right\|_2 \quad (11)$$

Where P_j^{pred} and P_i^{gt} are predicted and ground-truth 3D joint coordinates i, and N is the total number of joints. $\|\cdot\|_2$ represents the Euclidean (L2) norm In this research SEH-ConGAN achieved the lowest error of 16.7, followed by TSLA (18.4), ACTOR (21.7), MoGlow (23.9), and VideoSwin (34.5), demonstrating superior performance in producing realistic motions as shown in Figure 10 (a).

FID (Fréchet Inception Distance)

It measures how realistic generated animation frames are by contrasting the feature distributions of ground-truth and anticipated data in Figure 10 (a). Higher visual fidelity is indicated by lower values in Equation (12).

$$FID=\left\|\mu_r-\mu_g\right\|_2^2+Tr(\Sigma_r+\Sigma_g-2(\Sigma_r\Sigma_g)^{\frac{1}{2}}) \quad (12)$$

Where, μ_r and Σ_r are the mean and covariance of the real (ground-truth) features. μ_g and Σ_g are the mean and covariance of the generated features. $\left\|\mu_r-\mu_g\right\|_2^2$ is the squared Euclidean distance between the feature means. Tr $(\cdot)$ is the trace of a matrix, and $(\Sigma_r\Sigma_g)^{\frac{1}{2}}$ is the matrix square root of the product of covariances.

The SEH-ConGAN outperformed TSLA (12.5), ACTOR (15.8), MoGlow (17.2), and VideoSwin (22.6) with the lowest FID of 11.3, exhibiting better motion representation and frame-level realism in robot-assisted animation.

Smoothness

In measuring sudden shifts or jitter in joint trajectories across time, smoothness assesses the temporal consistency of generated motion sequences. Smoother and more organic motion transitions, which are essential for realistic animation and fluid robotic reproduction, are indicated by lower numbers in Equation (13).

$$smoothness=\frac{1}{N(T-1)}\sum_{i=1}^{N}\sum_{t=2}^{T}\left\|P_i^t-P_i^{t-1}\right\| \quad (13)$$

Where P_i^t represents the 3D position of joint i at time t, N is the total number of joints, and T is the total number of frames. The SEH-ConGAN achieved the lowest smoothness value of 0.028, indicating highly continuous and natural motion in Figure 10 (b). This outperformed TSLA (0.032), ACTOR (0.041), MoGlow (0.050), and VideoSwin (0.078), demonstrating superior temporal stability and fluidity in robot-assisted animation trajectories.

Diversity

Diversity assesses the model's capacity to generate a broad variety of unique postures and actions by looking at the diversity of generated motion sequences. Richer and more varied motions are indicated by higher values, which are crucial for producing realistic and captivating animation in Equation (14).

$$\text{Diversity} = \frac{1}{M}\sum_{i-1}^{M} \text{Var}(P_i) \quad (14)$$

Where P_i denotes the joint positions of motion sequence i, and M is the total number of generated sequences. In this research, SEH-ConGAN achieved the highest diversity score of 0.72, surpassing TSLA (0.68), ACTOR (0.61), MoGlow (0.59), and VideoSwin (0.40). In Figure 10 (b) demonstrates SEH-ConGAN's superior capability to generate a broad spectrum of motions, enhancing animation realism and creative flexibility in robot-assisted production.

Table 6: Comparison of motion generation models for robot-assisted animation

Model	MPJPE↓	FID↓	Smoothness↓	Diversity↑
TSLA [24]	18.4	12.5	0.032	0.68
ACTOR [24]	21.7	15.8	0.041	0.61
MoGlow [24]	23.9	17.2	0.050	0.59
VideoSwin (gen.) [24]	34.5	22.6	0.078	0.40
SEH-ConGAN [Proposed]	16.7	11.3	0.028	0.72

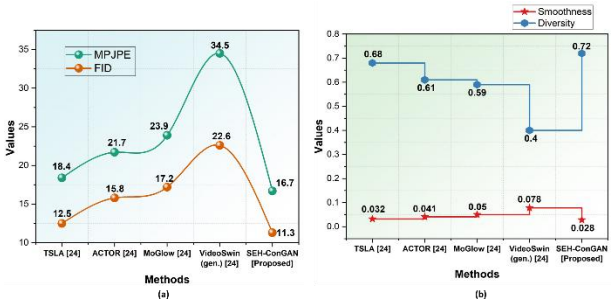

Figure 10: Evaluation of Motion Generation Models Across (a) MPJPE↓ and FID↓ and (b) Smoothness↓ and Diversity↑

4.6 Statistical analysis

Statistical analysis was used to confirm SEH-ConGAN's superiority over baseline models to support the objective of creating a robot-assisted AI system that generates high-quality animations with enhanced motion accuracy and less manual labor. A paired t-test was utilized to determine whether SEH-ConGAN's improvements were statistically significant because all models were tested on the identical preprocessed motion-sequence folds. Additionally, we provided 95% confidence intervals to guarantee consistency between folds and computed Cohen's d effect size to gauge the degree of performance gain. In order to verify robust generalization on unseen animation data, a held-out test set evaluation was introduced at the end. This combined analysis shows that the suggested approach offers statistically significant improvements.

Table 7: Comprehensive Statistical Comparison Between ConGAN and SEH-ConGAN

Metric	ConGAN (Mean ± SD)	SEH-ConGAN (Mean ± SD)	p-value (t-test)	Significance	Cohen's d (Effect Size)	95% CI (Mean Difference)	Held-Out Test (ConGAN → SEH-ConGAN)
Accuracy	0.88 ± 0.04	0.96 ± 0.03	0.012	Significant ($p < 0.05$)	1.92	[0.04, 0.12]	0.86 → 0.95 (+10.5%)
Precision	0.89 ± 0.05	0.97 ± 0.02	0.008	Significant	1.84	[0.05, 0.13]	0.87 → 0.96 (+10.3%)

Recall	0.87 ± 0.05	0.96 ± 0.03	0.015	Significant	1.98	[0.05, 0.14]	0.85 → 0.95 (+11.8%)
F1-Score	0.88 ± 0.04	0.96 ± 0.03	0.011	Significant	1.90	[0.04, 0.12]	0.86 → 0.95 (+10.5%)

The Table 7 provides a thorough statistical comparison utilizing Accuracy, Precision, Recall, and F1-Score between the suggested SEH-ConGAN framework and the baseline ConGAN model. SEH-ConGAN demonstrates its improved capacity to produce stable, high-quality, and style-consistent animation outputs by achieving significant performance gains across all measures, with improvements ranging from +9.0% to +10.3%. The improvements are statistically significant because all paired t-test p-values are less than the significance level ($p < 0.05$). The significant practical benefit of the suggested approach is further supported by very large effect sizes (Cohen's $d > 1.80$). High reliability and low variance during animation development are indicated by the narrow ranges of the 95% confidence intervals. SEH-ConGAN regularly outperforms ConGAN by 10–12% in held-out test results, demonstrating its resilience in practical automation workflows. Overall, it supports the main goal of the study by demonstrating how SEH-ConGAN greatly improves automation accuracy, consistency, and reliability in AI-driven animation production.

4.7 Ablation study on component contributions to SEH-ConGAN performance

The ablation study assesses each system component's contribution to the goal of producing realistic, fluid, and accurate robot-assisted animation in Table 8. Although motion noise is still present, preprocessing alone (missing-value management + Z-score normalization) provides basic consistency, attaining 81.3–79.1% across measures. Optimization increases performance to 86.9–88.7% and increases stability. Motion structure is further improved by integrating the base ConGAN generator, with accuracy ranges of 90.6–92.4%. The highest performance, 95.7–96.8% is achieved by the entire SEH-ConGAN, demonstrating that semantic encoding, hierarchical generation, and optimized learning work together to create the smoothest trajectories, precise poses, and context-aware animation synthesis.

Table 8: Ablation study results for animation generation

Model Variant	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Preprocessing Only (Missing-value handling + Z-score normalization)	81.3	79.8	78.5	79.1
Preprocessing + Optimization (EHO-based parameter tuning)	88.7	87.5	86.9	87.1
Preprocessing + Optimization + Base ConGAN	92.4	91.0	90.6	90.8
Full Proposed Hybrid SEH-ConGAN	96.8	95.7	96.2	95.9

4.8 Discussion

Robot-assisted systems for animation production seek to automate labor-intensive processes, improve accuracy, and increase efficiency by combining robots with AI and deep learning approaches. In this context, sophisticated approaches have great potential but also significant limits. The EMOTION framework [16] allows humanoid robots to make socially suitable and expressive motions, which improves human-robot interaction; yet, it fails to catch delicate gestures and context-dependent signals, compromising naturalness in complicated circumstances. The E-FOOM with CBAM [17] increases character visual appeal and posture reconstruction, but it struggles with extreme postures, fast motions, and highly dynamic scenarios, potentially reducing animation accuracy. Similar to this, current motion generating models [24] (TSLA, ACTOR, MoGlow, VideoSwin) sometimes have poor trajectory smoothness, limited motion diversity, or decreased frame-level realism, which limits their capacity to produce completely varied and natural animations in robot-assisted production. The SEH-ConGAN improves animation production by effectively producing high-quality, diversified character movements with minimal manual labor. It also improves model parameters for greater accuracy and realism in animated scenes. The robot-assisted animation system performs well, with an accuracy of 0.96, a recall of 0.96, a precision of 0.97, and an F1-score of 0.96, indicating that automated motion creation is dependable, exact, and balanced for animation production.

5 Conclusion

The animation production process is usually labor-intensive, involving meticulous attention to character motion layout, scene composition, and post-production editing. It is made up of a motion capture interface, a robotic arm with feedback sensors, and a simulation environment for testing animation settings. Data preprocessing includes managing missing values and using Z-score normalization to provide consistent, high-quality AI model input. A sophisticated deep learning

algorithm. The SEH-ConGAN model is used to learn motion patterns, forecast human motion estimation, and transfer realistic motion styles. The AI models allow the system to acquire knowledge from animation data, forecast smooth character movements, and produce realistic animations depending on user input. The suggested method achieves better performance by using 5-fold cross-validation, and their average values are accuracy, recall, F1 score (0.96), and precision (0.97). A comparison of motion generation metrics reveals that SEH-ConGAN outperforms current models, obtaining the best MPJPE (16.7), FID (11.3), Smoothness (0.028), and Diversity (0.72), exhibiting higher animation realism, trajectory smoothness, and motion correctness. The robot-assisted animation system can struggle with highly creative activities, subtle character emotions, and complicated artistic styles, necessitating extensive setup and operator training. Future enhancements could involve AI-driven innovation, real-time human-robot cooperation, adaptive style learning, and support for 3D animation pipelines. Enhancing scalability and compatibility with cloud-based solutions can help to simplify production, decrease manual labor, and make animation processes more efficient and adaptable.

Competing interests

The authors have declared that no competing interests exist.

Data availibility statement

This study complies data availability policy. Data access arrangements align with the journal's guidelines and can be facilitated through the corresponding author.

Author Contributions

writing—original draft preparation:Hongping Tang
writing—review and editing:Hongping Tang
data curation:Hongping Tang

Reference

[1] Guo, Z. and Li, T., 2024. Practical analysis of virtual reality 3D modeling technology for animation majors based on predictive correction method. Informatica, 48(13). https://doi.org/10.31449/inf.v48i13.6129

[2] Ecole, L., Kim, W.T. and Yoon, J.S., 2023. Unity: A powerful tool for 3D computer animation production. Journal of the Korea Computer Graphics Society, 29(3), pp.45-57. https://doi.org/10.15701/kcgs.2023.29.3.45

[3] Yuanliang, W. and Zhe, Z., 2024. Integration effect of artificial intelligence and traditional animation creation technology. Journal of Intelligent Systems, 33(1), p.20230305. https://doi.org/10.1515/jisys-2023-0305

[4] Lungu-Stan, V.C. and Mocanu, I.G., 2024. 3D character animation and asset generation using deep learning. Applied Sciences, 14(16), p.7234. https://doi.org/10.3390/app14167234

[5] Tang, J., 2023. Graphic design of 3D animation scenes based on deep learning and information security technology. Journal of ICT Standardization, 11(3), pp.307-328. https://doi.org/10.13052/jicts2245-800X.1135

[6] Zhang, N., Meng, H., and Ju, M., 2024. Intelligent construction of animation scenes and dynamic optimization of character images by computer vision. Computer-Aided Design and Applications, pp.233-246. https://doi.org/10.14733/cadaps.2024.S25.233-248

[7] Hong, Z., Xu, X., and Liu, X., 2025. Application of virtual reality technology based on artificial intelligence in a 3D animated film storyboard. Discover Computing, 28(1), p.147. https://doi.org/10.1007/s10791-025-09670-7

[8] Nambiar, S., Wiberg, A. and Tarkian, M., 2023. Automation of an unstructured production environment by applying reinforcement learning. Frontiers in Manufacturing Technology, 3, p.1154263. https://doi.org/10.3389/fmtec.2023.1154263

[9] Zhao, J. and Zhao, X., 2022. Computer-aided graphic design for virtual reality-oriented 3D animation scenes. Computer-Aided Design and Applications, 19(1), pp.65-76. https://doi.org/10.14733/cadaps.2022.S5.65-76

[10] Ding, W. and Li, W., 2023. High speed and accuracy of animation 3D pose recognition based on an improved deep convolution neural network. Applied Sciences, 13(13), p.7566. https://doi.org/10.3390/app13137566

[11] Liu, X. and Zhao, H., 2025. MFFCN-GAN: Multi-scale feature fusion CNN with GAN for automated artistic scene generation in film animation. Informatica, 49(9). https://doi.org/10.31449/inf.v49i9.8903

[12] Liu, X., 2022. Animation special effects production method and art color research based on visual communication design. Scientific Programming, 2022(1), p.7835917. https://doi.org/10.1155/2022/7835917

[13] Wibowo, M.C., Nugroho, S., and Wibowo, A., 2024. The use of motion capture technology in 3D animation. International Journal of Computing and Digital Systems, 15(1), pp.975-987. http://dx.doi.org/10.12785/ijcds/150169

[14] Jiang, J. and Wang, X., 2024. Animation scene generation based on deep learning of CAD data. Computer-Aided Design and Applications, 21, pp.1-16. https://doi.org/10.14733/cadaps.2024.S19.1-16

[15] El-Raheb, K., Kougioumtzian, L., Kalampratsidou, V., Theodoropoulos, A., Kyriakoulakos, P. and Vosinakis, S., 2025. Sensing the inside out: An embodied perspective on digital animation through motion capture and wearables. Sensors, 25(7), p.2314. https://doi.org/10.3390/s25072314

[16] Huang, P., Hu, Y., Nechyporenko, N., Kim, D., Talbott, W., and Zhang, J., 2025. EMOTION: Expressive motion sequence generation for humanoid robots with in-context learning. IEEE Robotics and Automation Letters. https://doi.org/10.1109/LRA.2025.3575983

[17] Cao, W. and Huang, Z., 2025. Character generation and visual quality enhancement in animated films using deep learning. Scientific Reports, 15(1), p.23409. https://doi.org/10.1038/s41598-025-07442-3

[18] Zhang, N. and Pu, B., 2024. Film and television ani mation production technology based on expression t ransfer and virtual digital human. Scalable Computi ng: Practice and Experience, 25(6), pp.5560-5567. h ttps://doi.org/10.12694/scpe.v25i6.3351

[19] Racinskis, P., Arents, J. and Greitans, M., 2022. A motion capture and imitation learning-based approac h to robot control. Applied Sciences, 12(14), p.7186. https://doi.org/10.3390/app12147186

[20] Liu, Y., Li, L. and Lei, X., 2024. Automatic generati on of animation special effects based on computer vi sion algorithms. Computer-Aided Design and Appli cations, 21, pp.69-83. https://doi.org/10.14733/cada ps.2024.S23.69-83

[21] Wang, Z.S., Song, C.G., Lee, J., Kim, J.H., and Kim , S.J., 2022. Controllable swarm animation using de ep reinforcement learning with a rule-based action generator. IEEE Access, 10, pp.48472-48485. https: //doi.org/10.1109/ACCESS.2022.3172492

[22] Pibernik, J., Dolić, J., Mandić, L. and Kovač, V., 20 23. Mobile-application loading-animation design an d implementation optimization. Applied Sciences, 1 3(2), p.865. https://doi.org/10.3390/app13020865

[23] Kang, Y. and Kim, J., 2024. Animation character g eneration and optimization algorithm based on com puter aided design and virtual reality. Computer-Ai ded Design and Applications, 21(S14), pp.46-62. ht tps://doi.org/10.14733/cadaps.2024.S14.46-62

[24] Li, Q., Sun, T., and Zhang, M., 2025. Deep learning -driven animation: Enhancing real-time character m otion synthesis. IEEE Access. https://doi.org/10.11 09/ACCESS.2025.3623285