

A Probabilistic Transformer for Seed-Conditioned 2D and 3D Dance Motion Continuation

Fiona Yan¹

Received May 17, 2026

Accepted August 30, 2026

Electronic access October 15, 2026

We ask whether a seed-conditioned generative model can extend a short dance phrase into a longer continuation that stays temporally smooth and keeps a readable skeleton. The dancer or animator chooses the seed, reviews, revises, or rejects the output, and retains control of the composition, while the system acts as a drafting tool to speed up creative work. We design a probabilistic Transformer encoder for 2D or 3D keypoint poses and train it on the AIST++ dataset of ten dance genres. Our model outputs relative changes in pose between frames, with associated uncertainties. We hypothesize that this alleviates mean-pose freezing and encourages pose continuity during autoregressive rollout. As a post-processing step, we rescale the lengths of body segments and reposition the head with respect to the shoulders. These modifications may limit the poses the model can express, a trade we accept to prioritize usability over unbounded, exaggerated motion. Our exploratory results, including loss curves, Gaussian Negative Log-Likelihood (NLL) on held-out data, and visual inspection, indicate that the generated drafts move continuously, retain the local rhythm and spatial structure of the seed, and keep a connected skeleton, although held-out fit varies across runs. We have not included a user study or a direct comparison with a Transformer that predicts absolute poses, so we do not claim that our model performs better than alternatives. We believe this work is a first technical step toward such models in creative workflows, supplying readable transition drafts while human agency and authorship remain with the choreographer.

Keywords: AI Dance Choreography, 2D/3D Human Motion Synthesis, Motion Continuation, Probabilistic Transformer, Autoregressive Motion Prediction, AIST++

Introduction

Background

Our approach to deep learning-based motion generation uses recorded dancing as the source of truth for motion structure, and seeks to avoid encoding any rules beyond those which a deep learning model can learn from data. Dance offers a concrete and measurable setting for this idea. Given a short seed motion, defined by a set of skeletal keypoints over time instead of video pixels, we seek to generate continuations of that motion that are statistically and visually plausible. We believe such a system could prove helpful to creative practitioners, for instance to suggest motion for a given context or to save some time during an animation task. Although we do not study the use of our system by dancers and choreographers in this work, such a user study would be interesting future work.

We propose a tool which a human practitioner can use to suggest possible dance moves, and which learns its patterns from recorded movement instead of the hand-written rules of the computational choreography tools that have shaped dance composition for decades¹. Plausibility will always be a re-

quirement of such a tool, although it does not by itself demonstrate usefulness, and we discuss the statistical and visual plausibility that a system like this may aspire to, and the ways in which our own approach is limited.

Although the neural network we propose does learn certain relationships in the body, for instance that the shoulders may move when the head does, it does not encode physical mechanics of the body such as balance or the forces required to perform a certain dance move. The network's primary representation is one of keypoints in time, and while the effects of physical factors such as force and friction leave partial traces in this data, the system does not model them directly, nor can it reason explicitly about concepts such as contact with the floor. For this reason, we hope a system like ours will still be used with a human animator in the loop, as the final critic and editor of the results. Similarly, we do not attempt to model or influence higher-level attributes of a dance, such as its structure, musicality, or phrasing. Such qualities will still be up to the choreographer and dancer, who can also control the artistic goal and intent of a given dance. Ultimately, such a tool could let the user explore a wider set of possibilities than they may have otherwise considered, even if many of those options are still less than ideal.

¹ St. Robert Catholic High School, Canada

Research Question and Hypotheses

Our central research question is stated below.

Can an autoregressive probabilistic Transformer trained on AIST++ keypoints extend a short dance seed into a longer 2D or 3D motion sequence while preserving temporal continuity and skeletal structure?

The alternative hypothesis of this study is that modeling pose transitions with a probabilistic model leads to more smooth and structurally coherent human motions, compared to modeling absolute poses. The null hypothesis is that it does not.

We want to note here that there is no ablation study in this version of the project and that no baseline is being compared with the transition, or delta, model. The missing baseline is an absolute-pose model, which we describe as future work in the Discussion section. For this reason, we cannot verify this hypothesis in this article and leave it to a controlled study, described in the Discussion section.

For this reason, the results of this work should be considered exploratory. As described in the Discussion, there are several factors that can potentially contribute to the apparent smoothness of the output, including the skeleton correction technique, the Gaussian likelihood, the sampling procedure and the fixed-length context window, and the use of the Transformer architecture itself. We do present the negative log-likelihood of our delta model on held-out data, and give some qualitative results and loss curves, as evidence for its behavior.

Literature Review

Given the sequential nature of motion data, many early methods on neural human dynamics², motion, and dance generation adopt a recurrent network as the backbone. Fragkiadaki et al. introduced the encoder-recurrent-decoder architecture for human dynamics², and recurrent generation has been adopted in dance generation systems such as chor-rnn, a generative LSTM system for choreography³. While these models are effective on short-term prediction tasks, their autoregressive rollouts drift into unrealistic poses due to the error accumulation². More specifically, given a context motion, the model feeds each generated frame back as input to predict the next frame. This approach is referred to as autoregressive, as the process predicts one frame at a time to generate motions. The most common recurrent models are the long short-term memory network (LSTM) and the gated recurrent unit (GRU), which introduce multiplicative gating to control the flow of information to mitigate the vanishing gradient problem. As these models are rolled out beyond their training horizon, they produce unnatural poses, freezing or jittering. To mitigate this issue, Huang et al. use a curriculum training schedule for long-term dance generation⁴. Over the course of training, the curriculum gradually replaces the ground-truth inputs with the

predicted output of the model to better handle the error accumulation.

When it comes to the design of the target, Martinez et al. showed that contemporary recurrent predictors could be matched by a zero-velocity baseline that repeats the previous pose⁵. They found better results with a residual connection that predicts the velocity rather than the pose. A similar idea was applied to rotation prediction in QuaterNet⁶, which also models change rather than the pose itself. Following this line of reasoning, we choose to use a delta target for the task of dance continuation.

While the early methods were based on recurrent neural network (RNN) architectures, more recent research has shifted towards Transformer-based solutions⁷. A major design difference between the Transformer models and RNN-based ones is the attention mechanism: rather than relying on the hidden state of the last frame and passing on this state, the Transformer can directly attend to each of the frames in its context window. Attention does not make the context unlimited, but within the available window it does compare all frames directly. We expect an attention mechanism to be particularly useful in modeling dance continuation, where one may need to account for motion patterns that repeat over time or limbs that may be delayed in response to motion. While we still perform prediction in a stepwise manner, we apply a causal mask to ensure the model does not receive future ground-truth frames. The value of attention-based 3D dance generation conditioned on music and an initial motion seed was shown with the introduction of the Full-Attention Cross-modal Transformer (FACT) model with AIST++^{8,9}. Other attention-based systems, such as the spatio-temporal Transformer of Aksan et al., 2CH-TR and Human MotionFormer, also model complex human motion^{10–12}.

Music-conditioned dance generation is now a mature sub-field. Lee et al. decomposed dance into reusable units and noticed that dance is inherently multimodal, since there are several following movements of a pose that are equally plausible at any moment¹³. Bailando quantized dance into a learned codebook, a fixed vocabulary of reusable movement units, and composed the units with an actor-critic GPT rewarded for beat alignment¹⁴. EDGE uses a Transformer diffusion model with a strong music encoder, supports editing operations such as dance continuation, and was validated through a large-scale user study¹⁵. As opposed to these music-conditioned systems, we isolate motion-to-motion continuation: we probe what can be produced from the internal dynamics of the seed itself, without any audio or text conditioning, using a much smaller model.

Several generative families are still relevant. Variational Autoencoders (VAEs) learn a structured latent space that enables controllable sampling. In layman's terms, a latent space is a compressed internal coordinate system learned by the

model in which similar motions are located close to each other¹⁶. ACTOR integrates a Transformer with a VAE for action-conditioned synthesis¹⁷. Generative Adversarial Networks (GANs) and Denoising Diffusion Probabilistic Models (DDPMs) are also popular^{18,19}, and diffusion has emerged as a leading approach for general human motion generation²⁰. Diffusion models can produce high quality samples but iterative denoising requires many inference steps per output. Latency matters for an interactive choreography tool, so we use a probabilistic autoregressive model that does a single forward pass per generated step. There is a strong precedent for likelihood-trained autoregressive motion models: MoGlow synthesizes motion from an explicit probability distribution trained with maximum likelihood²¹, Transflower uses probabilistic autoregressive modeling specifically for dance²², and DLow demonstrates the importance of representing multiple plausible futures, and sampling diverse ones, for human motion prediction²³.

Problem Difficulty

The goal of our experiments is to explore various design choices to make dance generation that is both plausible and kinematically valid. We now discuss the challenges specific to long-term human motion generation that motivate these choices. Our first concern is to predict the full distribution over future poses instead of only the mean pose. Given the same initial pose, there are many possible actions a dancer could take, including stopping, turning, shifting their weight, extending a limb, and continuing in another direction. In other words, the future is multimodal. A deterministic model trained via mean squared error would predict some mean pose averaging these futures, and in pose space this tends to be a centered pose that results in frozen motion. Predicting the distribution is a common strategy to avoid these freezing artifacts, and is particularly critical in dance, where the generated motion would greatly suffer if the dancer unexpectedly came to rest, unless the target dance deliberately contains stillness.

Our second concern is to predict kinematically valid skeletons, meaning stable body-segment lengths and a head anchored to the shoulder line. If we predict each body part in isolation, say the wrist, elbow, shoulder, head, and torso, we are not necessarily going to keep the same distance between connected joints across the sequence, leading to rubbery limbs and to the head decoupling from the body, especially during fast movements.

We frame the generated motion as a possible drafting tool for a choreographer, and motion phrases with decoupled heads or variable arm lengths would not be helpful. With these goals in mind, we next discuss the design of our model and the trade-offs between these objectives.

Objectives

Our goal is to learn a probabilistic model for dance continuation: given a short input sequence of 2D or 3D keypoint observations, predict a distribution over possible next steps, that is, changes in pose from one frame to the next. We propose a seed-conditioned Transformer model to address this task and train separate models for 2D and 3D motion sequences on AIST++ keypoint trajectories using the same architecture. We predict transitions, rather than absolute poses, to reduce freezing and damping artifacts. Transitions maintain local momentum and aid in continuation, while the poses recovered from them are easy to visualize. The distributional nature of the output can be used to capture uncertainty in possible continuations, which is crucial in applications such as dance, where several continuations are plausible from the same seed. We propose a post-processing step to correct common generation errors, such as the head drifting away from the shoulders and incorrect limb lengths. The final outputs are sampled continuations that are geometrically consistent. Our evaluation is conducted using held-out Gaussian NLL, loss curves, and qualitative inspection.

We do not claim that our model is creative. Rather, our goal is to explore a workflow where humans could use our system's output to explore many plausible dance continuations and choose which are most meaningful, as well as refine, curate, and iterate until they are happy with the output. This study explores the plausibility of such a system's output, and it includes no comparison with baselines, so we do not claim ours is better than other models. Future work includes further qualitative studies, such as user studies, and comparisons with other models that would indicate its utility to choreographers.

Scope and Limitations

Our model is trained for solo dance motion, and not group dancing, and only generates a type of motion that can be expressed with keypoints. We can learn from 2D and 3D AIST++ data, but we do not know how well the model generalizes beyond the styles in which the professional dancers in AIST++ perform. Our current version does not condition on music, text, scene, or performer identity. The output is therefore a continuation from a seed, and generating full choreography from scratch is out of scope.

Moreover, we do not train a baseline that predicts absolute pose to directly test the delta hypothesis, and our NLL evaluation on held-out data uses teacher forcing, conditioning on real previous frames, which does not reflect the performance of our network during long rollouts. The current model relies on our skeletal correction to prevent extreme, physically impossible cases where joints separate, but does not model center of mass or force. Therefore, our model can slide its feet instead of

maintaining floor contact, since it does not represent the force applied by muscles to control the bones.

In practice, enforcing valid limb lengths can limit the range of motions our network can output, as it may not be able to reproduce the exaggerated arm trajectories of a Waacking dance. However, we believe that it is better to make a conservatively readable skeleton at the cost of some stylistic motions, as opposed to an unconstrained skeleton that is more expressive but may break down in fast or extreme motions. As discussed in the Discussion section, there may be benefits to building more advanced constraints into the model, such as modeling foot-floor contact, but we limit our scope to ensure that our model can at least draft reasonable dance motions, rather than attempting state-of-the-art biomechanics. Finally, our evaluation protocol relies on offline metrics and researcher inspection, rather than a formal user study with choreographers to provide subjective feedback on the usefulness of generated results.

Methods

Research Design

This work consists of a method contribution and an evaluation. We build a Transformer-based model for seed-conditioned motion continuation in 2D and 3D. For evaluation we consider the Gaussian NLL on a held-out dataset, as well as qualitative evaluation of the generated sequences, visualized as skeleton animations, where we check for discontinuities, freezes, shrinking or growing of body part lengths, and the separation of head and body. We deem both essential because the NLL does not necessarily capture dance-plausible movements, while qualitative assessment does not provide quantitative insights.

Data Collection

We use the publicly available AIST++ dataset⁸, which consists of 1,408 dance sequences with synchronized music, multi-view video, and reconstructed 3D body motion, of which we use only the body keypoints. AIST++ has been derived from the AIST Dance Video Database⁹ and captures different genres of dance and a wide range of motion dynamics, from intense floor work such as Break and Krump to dances mostly involving upright poses such as Ballet Jazz, all of which a single architecture must accommodate. To capture intentional dances and not incidental movements, the 3D body motion has been captured from 30 professional dancers in controlled conditions. The considered genres are: Break, Pop, Lock, Middle Hip-hop, LA-style Hip-hop, House, Waack, Krump, Street Jazz, and Ballet Jazz.

For modeling body motions we consider the 17 body keypoints in COCO format^{24,25}, namely the nose, eyes, ears, shoulders, elbows, wrists, hips, knees, and ankles, and not the complete SMPL body model or input videos. Given a sequence of 2D or 3D body keypoints we get a sequence of matrices X_t , one for each frame t , whose i -th row describes the location of the i -th body joint in the t -th frame. The locations consist of image plane coordinates and confidence scores in 2D, or Cartesian coordinates in 3D. We drop the confidence scores so that each 2D pose has 34 values per frame, and each 3D pose has 51 values per frame. Table 1 details the joint indices, whose ordering is used throughout this work.

Data Preprocessing

The raw data for our experiments are keypoint sequences extracted from dance videos. There are several ways we can process this data before using it for our experiments. First, we consider a 2D setting with only the image-plane keypoints and a 3D setting with the 3D coordinates. The 2D keypoints are always measured from a single viewpoint. For 3D data, we use the global 3D coordinates, instead of projections onto different views.

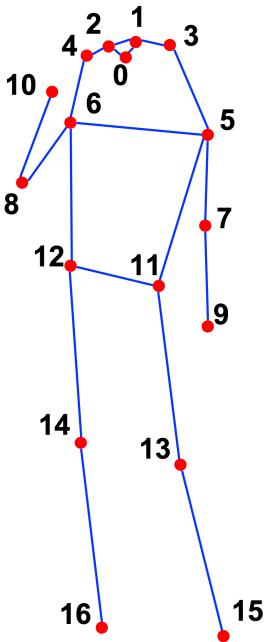
We also exclude the confidence scores, which are present as the third dimension in the 2D data, because we want our input signal to be a purely geometric quantity, with a fixed meaning in each dimension. Otherwise, mixing a detection-quality signal into the same vector as coordinates would ask the network to model two unrelated kinds of information at once. This choice limits robustness to occlusion and other situations where there may not be useful information about the pose in the input images, but is an acceptable limitation to make, given the studio-quality data in AIST++.

We remove any sequences with non-finite values, rather than attempt to fill the data in. This filter removes 20.2% of the 1408 3D sequence files, and 2.2% of the 1510 2D sequence files, where the 2D release contains more files because some recordings do not have 3D reconstructions available for download. We normalize the remaining data by the mean and standard deviation in the train split, to ensure consistent magnitudes between different axes and make convergence more rapid.

We partition the sequences into windows of length $T = 30$ frames. We take frames 1 through $T - 1$ of each window as inputs to predict the next-frame delta, which is the amount of change in the pose from one frame to the next. We use a windowed dataset to focus the model on learning short, 0.5-second dance phrases, given the dataset's frame rate of 60 frames per second, while still yielding a usable set of non-overlapping windows from a single recorded sequence. We did not systematically vary the window length, and the Results suggest that a longer context could help some styles.

Table 1 COCO keypoint index and the corresponding body landmark. The numbered joints define the order of the coordinate matrix supplied to the model, so the same index-to-body-part mapping is used during preprocessing, training, generation, and visualization. The numbered skeleton below the table shows the same mapping drawn on the COCO topology; the diagram is reproduced from the MMPose documentation²⁶.

Keypoint Index	Position on Human Body
0	Nose
1	Left Eye
2	Right Eye
3	Left Ear
4	Right Ear
5	Left Shoulder
6	Right Shoulder
7	Left Elbow
8	Right Elbow
9	Left Wrist
10	Right Wrist
11	Left Hip
12	Right Hip
13	Left Knee
14	Right Knee
15	Left Ankle
16	Right Ankle



We also split the dataset into training, validation, and test datasets, which we describe in more detail below, to get a sense of the quality of generalization of our models. Each run draws on a single source sequence, such as Lock (3D) sequence A, which we divide into training, validation, and test

datasets. Each sequence is partitioned temporally into contiguous segments of windows. We generally split our data 70% training, 20% validation, and 10% test. Because counts are rounded down to whole windows, an experiment with 24 windows has 16, 4, and 4 training, validation, and test windows, or 67%, 17%, and 17%. The dataset uses a stride equal to the window size, to ensure that the data in the windows do not overlap. In the individual sequence experiments, we split Lock (3D) sequence A into 41 training, 11 validation, and 7 test windows; Lock (3D) sequence B and Break (3D) each into 16, 4, and 4; and Street Jazz (2D) and Ballet Jazz (2D) each into 14, 4, and 3. We list these individual datasets in Table 3. This dataset splitting methodology has two downsides. First, the samples are quite few in number, which means our NLL estimation for each split has a high variance. Second, we are splitting within sequences, and not by dancer or by dance, so we are only looking at generalization ability to unseen portions of dance sequences, instead of truly unseen dancers or dances. A sequence-level split, ideally with a dancer held out, is the appropriate stronger test for future work.

The delta prediction task focuses the model on capturing the direction and speed of motion, instead of the pose itself. It also helps avoid a regression to the mean, where the model would collapse to a central mean pose instead of the true pose. Also, when the future is multimodal, averaging over the plausible futures damps the resulting motion far less for a delta target than for an absolute-pose target, which damps towards a single mean pose. We do not use the proposed geometric correction to the model outputs for more skeletal consistency during training. This correction is helpful when sampling continuations, though, so we use it in the Generation section.

Data Visualization

Figure 1 visualizes the poses after preprocessing in order to check that they contain all of the joints and that they are connected in the correct way. Each pose in the figure contains all 17 joints and has no missing or disconnected parts. We thus conclude that our 2D preprocessing step preserves the skeleton structure. We can see in this plot that the COCO topology is preserved during the data standardization process and removal of the confidence channel. These poses cover the extremes, such as the very vertical Ballet Jazz poses and the very low Break and Krump poses, and all of these genres should be representable by the same model architecture.

Figure 2 plots similar poses, but in 3D, and confirms that the pipeline preserves connected body structure and genre-level pose variation across all ten styles before any temporal modeling. While any slight coordinate errors are barely visible in Figure 1 and Figure 2, they would become more apparent when animating these frames in sequence. This difference highlights the importance of the anatomical correction step.

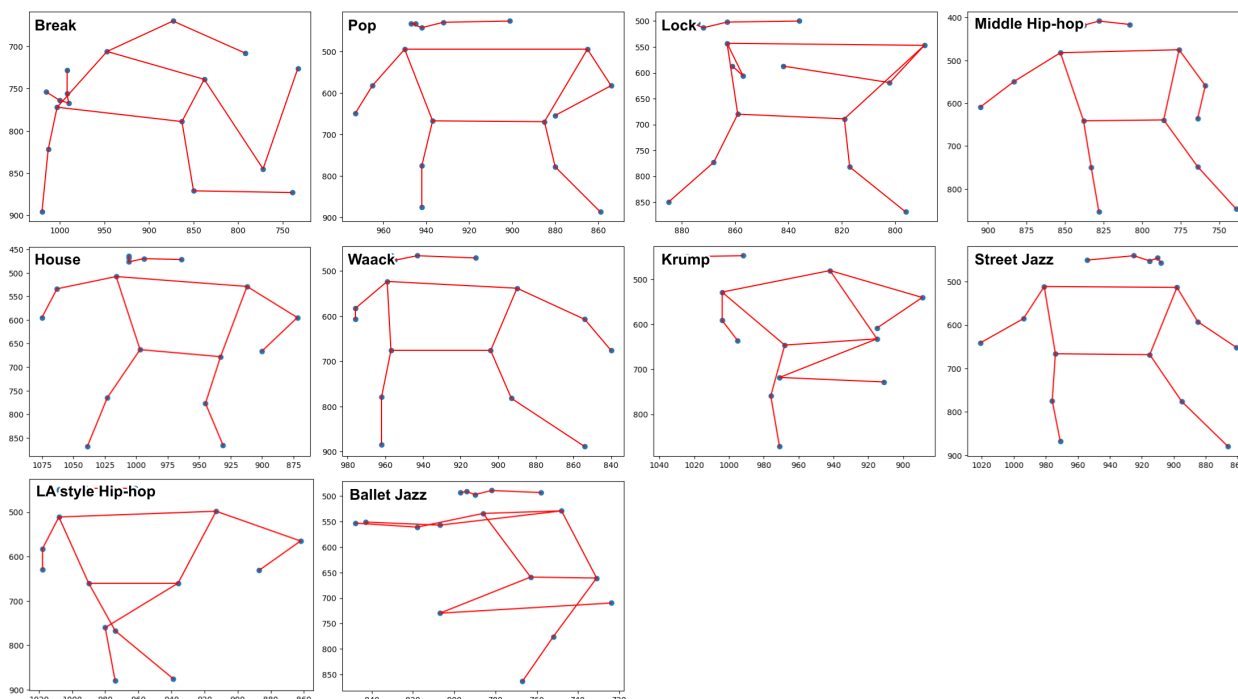


Figure. 1 Representative 2D COCO keypoint frames from the ten AIST++ dance genres. Blue points mark joints and red lines indicate skeletal connections. Comparing panels shows the postural spread the model must fit: Break and Krump sit in compact, low crouches, while Ballet Jazz holds an upright, extended line. Each panel is a single example frame and does not represent the full variety of poses within its genre.

Architecture

For training purposes, the motion continuation model is applied over a window of time, where each pose p_1 through p_t corresponds to a frame within the time window. We use the following definition for the input and target

$$x = \{p_{1:t-1}\}$$

$$y = \{\delta_{1:t-1} = p_{2:t} - p_{1:t-1}\}$$

Here $\delta_{1:t-1}$ denotes the sequence of changes between subsequent frames.

Predicting the change between frames leads to more stable training, and helps avoid the tendency of a model trained with mean squared error to collapse toward a static mean pose, since there is less variation in the targets and they are centered around zero.

Each pose vector is passed to a linear embedding layer, resulting in a 128 dimensional vector representation, and is then summed elementwise with a positional embedding that is learned jointly during training, in place of a fixed sinusoidal encoding. This sequence of vectors is passed through a Transformer encoder network with three layers, where each layer has multi-head self-attention with four heads, followed by a position-wise feed-forward network with a hidden layer size

of 512 dimensions. ReLU activations are used in the feed-forward layer, and all layers use a dropout rate of 0.1. Multi-head self-attention is used to learn the relationships between frames within the window. Each attention head projects the frames into query, key, and value vectors and weights frames by their query-key similarity, and different attention heads are able to learn different relationships in parallel. A causal mask is applied during self-attention to train the model to predict future poses autoregressively. This is done by masking out all positions above the diagonal of the attention matrix, such that the pose for frame i can attend only to the current and previous frames, not to future frames.

The output vectors from the final layer of the encoder network are passed to linear output layers which are responsible for outputting the next change δ_t . Since several future motions are plausible from the same past, we train two separate output layers in parallel. One predicts the mean μ_t of a distribution for each joint coordinate of the next change, and the other predicts a corresponding log standard deviation s_t . This log standard deviation is clamped to a bounded range for numerical stability. We model the distribution over the future as a Gaussian, the standard bell-curve distribution, so the model is predicting a most likely change and how spread out the plausible changes are around it,

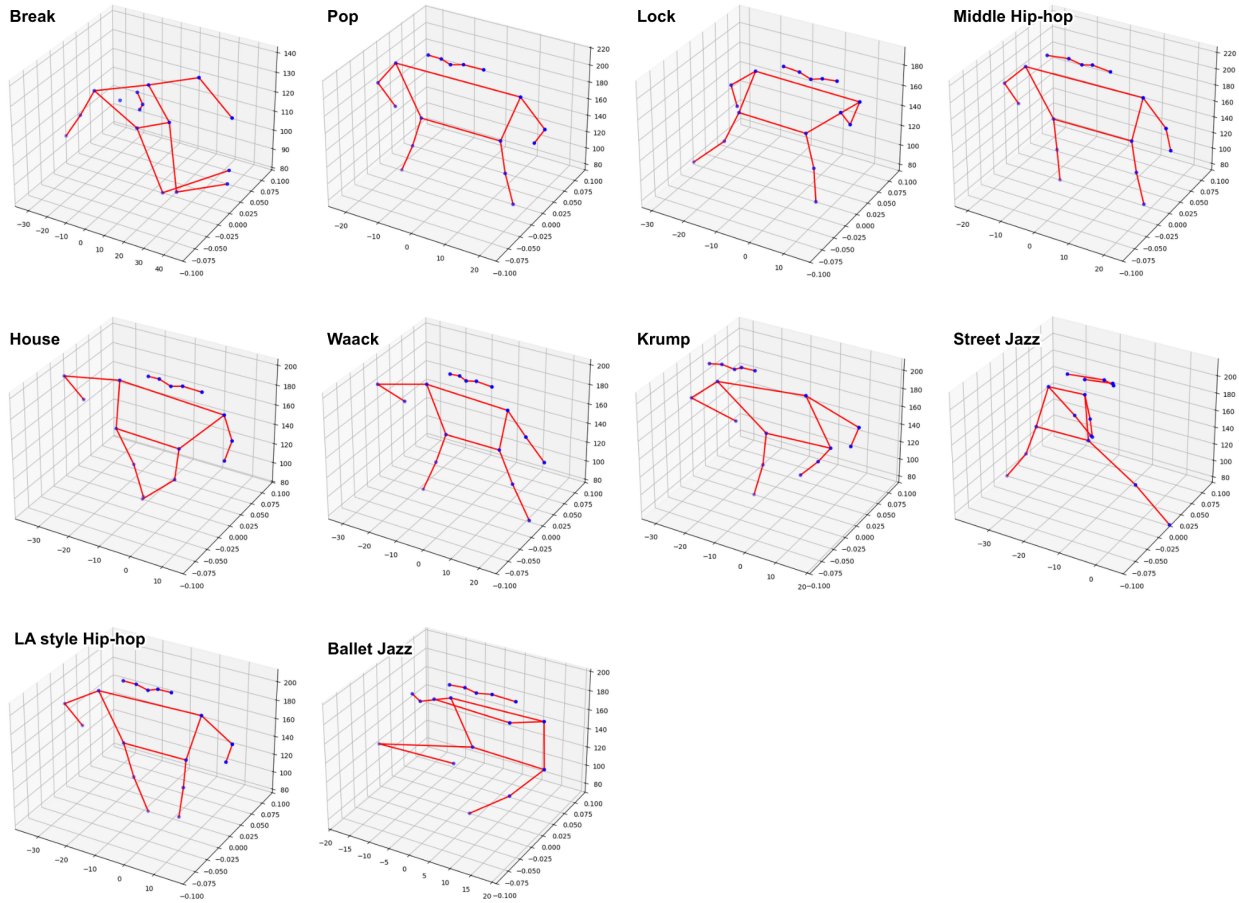


Figure. 2 Representative 3D keypoint visualizations from the same ten genres, used to confirm that the parsed joint coordinates preserve skeletal structure across styles. As in Figure 1, note the contrast between grounded genres such as Break and the extended carriage of Ballet Jazz; these are single example frames only.

$$\delta_t \sim N(\mu_t, \sigma_t^2)$$

and use its negative log-likelihood as the loss during training:

$$\text{NLL}(\mu, \sigma^2, y) = \frac{1}{2} [\log(\max(\sigma^2, \epsilon)) + (\mu - y)^2 / \max(\sigma^2, \epsilon)]$$

where σ is the clamped standard deviation, ϵ is a small value to avoid taking the logarithm of zero, and μ and y are calculated as above. In plain terms, the NLL measures how surprised the model is by the true next change, so a lower value means the model assigned more probability to what actually happened. The constant term $\frac{1}{2} \log(2\pi)$ is left out of the expression, which means the true NLL of a distribution is this value plus approximately 0.92 for each joint coordinate, and is one more reason the reported values should be read comparatively. We report the averaged loss over the joint coordinates

and the timesteps within the window. Using a distribution enables the model to learn regions of high and low variability rather than simply minimizing an error metric symmetrically for over- and under-estimates of future poses. Training with such symmetric error measures tends to produce the undesired result of learning a mean pose. A distributional output with a corresponding loss is intended to produce autoregressive roll-outs that do not freeze. Table 2 lists the architecture parameters.

Training Parameters

Our models are trained using the AdamW²⁷ optimizer with a learning rate of 3e-4. The batch size is 8 and we train for up to 200 epochs, after which validation metrics start to degrade. Batches are shuffled during training and fixed in validation and testing.

We set a max norm of 1.0 for gradient-norm clipping, as a

Table 2 Model architecture parameters.

Parameter	Value
d_model	128
num_heads	4
num_layers	3
dim_feedforward	512
dropout	0.1
activation	relu

batch with unusually large coordinate changes can otherwise produce an update that destabilizes learning, especially in the attention layers. Results in this paper are reproducible, as we fix random seeds for Python, NumPy and PyTorch, and ensure that only deterministic algorithms are called.

Generation

For autoregressive inference, a seed sequence for the model is given, which can be seen as a short sequence of input that allows the model to generate a continuation. To match the window size of the model during training, we use a seed length of 30 frames, and generate a rollout with 100 frames. At 60 fps, this rollout corresponds to about 1.7 s of motion. We report this length because many models look plausible over short rollouts and degrade over long ones.

In a standard autoregressive model, the next frame is inferred from the previous ones. In our case, we instead predict the delta vector from the last frame in the model’s context to the next frame, meaning that, for example, when predicting the position p_t , the input sequence will be $\text{model}(p_{t-n}, \dots, p_{t-1}) \rightarrow (\mu_t, \sigma_t)$. Here, instead of a full pose, a pair of mean μ and standard deviation σ of the distribution for the next delta vector δ_t is predicted. The actual delta to apply is then sampled from the predicted distribution

$$\hat{\delta}_t = \mu_t + \varepsilon \sigma_t,$$

where $\varepsilon \sim N(0, I)$ is Gaussian noise, and the pose is found using numerical integration

$$p_t = \hat{\delta}_t + p_{t-1}.$$

Instead of taking the full history as input for the prediction, we take the most recent 10 frames as context. Contexts of this length also occur inside every training window because of the causal mask, so the model has been trained to predict from short contexts as well.

Since sampling from the predictive distribution leads to a variety of possible outputs, the same seed can produce different but plausible continuations, which an exploration tool requires. Furthermore, the standard deviation is capped at 0.05, meaning that the predictive distribution’s standard deviation is

at most 0.05 standardized units. This cap ensures that large jumps are unlikely to happen, and narrows the sampling distribution relative to the one the NLL evaluates, so the diversity across samples from one seed is more limited.

Instead of predicting poses directly, we predict the deltas between two subsequent poses. Because each predicted pose becomes part of the input for the next step, coordinate errors can still compound over a long rollout, but the delta limits errors locally at each step, so absolute position cannot silently drift within a single prediction.

Generating a sequence of 100 poses takes 1.1 to 1.6 s on a single Tesla T4 GPU, since each generation step is a single forward pass, which supports the interactivity motivation stated in the Literature Review.

To ensure readability, and as the model is not intended to model the actual dynamics of the body, post-processing is used to enforce constraints on body geometry. The four steps are the following. First, we compute each body part length, defined as the median Euclidean distance between two connected joints, using the seed sequence. Second, we center the sequence on the hips by taking the midpoint of the two hips on each frame and subtracting it from all other points. Third, we iteratively rescale the body part lengths to those computed in the first step using five iteration steps. Fourth, we correct for head drift by finding the median head position in relation to the shoulders in the seed sequence and re-setting the head key-points according to these relations. The correction suppresses visibly impossible skeletons while leaving the underlying motion to the probabilistic model.

For visualization of the results, the generated 2D or 3D sequences of poses can be plotted as animations using FuncAnimation²⁸.

To assess the quality of the sequences, we conduct a qualitative analysis. For each model, we sample between three and six rollouts of 100 frames. These are then viewed as animations in 2D and 3D. When doing so, we analyze the videos using a fixed checklist: whether the motions are reasonably continuous, whether the subject sometimes freezes, whether body parts drift, and whether the head is separated from the shoulders. The inspection was researcher-led, and we did not use blinded raters or a scoring rubric beyond this checklist, which we note as a limitation.

Ethical Considerations

Our work collected no primary data and did not involve human or animal subjects, thus IRB approval, informed consent, and confidentiality are not applicable. All experiments were conducted using the public AIST++ dataset, which was recorded from professional dancers for research purposes and released for academic use, and we use it within that scope and cite it accordingly.

This tool is designed for drafting only. The dances generated by the model are to be reviewed by humans, and our tool aims to assist, rather than replace, choreographers. The intent and final decision lie with humans. Also, the generations should only be understood as representing the styles of dance recorded in AIST++, and should not be taken as representing dance practice beyond the genres and performers in that dataset.

Results

We evaluate our model with likelihood estimation and visual inspection. Given the previous frames, likelihood is an indicator of how much probability density the model's learned distribution assigns to real pose transitions in unseen data. However, it is not necessarily reflective of the visual quality of generations. Conversely, visual inspection offers a qualitative evaluation but is not indicative of the predictive performance. It is thus informative to conduct both types of evaluation. We estimate the held-out negative log-likelihood (NLL). Given a sequence, we evaluate the model with teacher forcing and report the NLL of the next transition given the actual past frames. This metric evaluates our model as a one-step predictor rather than the quality of autoregressive samples. For visual inspection, we present the generations and ask whether the generated skeletons are readable as dances over time. We leave the evaluation of sequence-level generation using roll-out to future work, including the per-seed sample diversity and common artifacts in learned models such as foot sliding and bone-length error over generated sequences.

Quantitative Generative Performance

Table 3 lists loss values for five runs over four genres and over two keypoint modalities. It should not be interpreted as a ranking, but merely as a sample of results for illustration; several other genres were trained but are omitted for space. All the runs in Table 3 share the same loss function, the same standardization and the same choice of window length, so their loss values are comparable. The NLL of a continuous density can take any real value, as there is no zero lower bound, so the metric is informative relative to itself. In particular, the loss used to train the networks can take on strongly negative values when training goes well. A strongly negative value is better than one closer to zero, as a negative value means that the model has placed large probability mass, with low variance, on the true target. Closer to zero, or above it, the probability mass is more diffusely distributed.

In general, the loss curves show significant movement in the first few epochs, and then make much slower progress towards eventual stability, as expected for this class of models. However, the fit to held-out data often falls substantially short

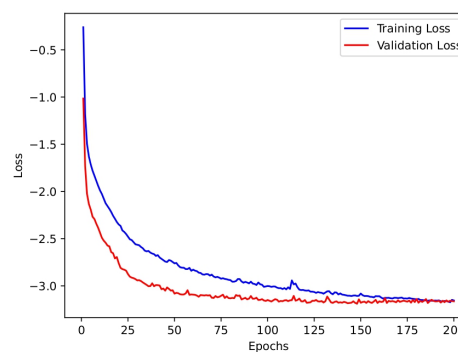


Figure. 3 Training and validation NLL for 2D Street Jazz over 200 epochs. Both curves fall quickly and then flatten, indicating that the model settles into a consistent transition distribution before training ends.

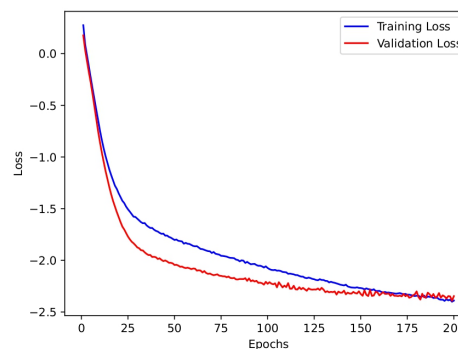


Figure. 4 Training and validation NLL for 3D Break over 200 epochs. The validation curve stays close to the training curve late in training, which suggests stable generalization for this run.

of that in the training set. In some runs validation loss closely tracks training loss, while in others it falls well short of it, so the informative comparison is between training fit and held-out fit generally. The validation NLL may be slightly worse than the test NLL, as in one of the runs in Table 3, merely because this is a noisy estimator in our setup: that run has only 11 windows for validation and 7 for testing, so orderings between adjacent columns should not be over-read.

We can infer some things from Table 3. In the two Lock runs, we trained on two distinct dances, and the columns have quite different values, so the variance is not captured by genre alone but by specific dances and splits within them. We cannot tell whether a 30-frame window captures enough to make a good model for all genres. We cannot say whether 2D or 3D is easier, or whether Street Jazz is easier than Ballet Jazz, as we ran each only once, with one seed; several runs per genre

Table 3 Training, validation, and test NLL for a sample of trained runs across both keypoint modalities. The two Lock rows are separate models trained on different source dances of the same style. Additional genres were trained but are omitted for space; the entries are illustrative and not a ranking.

Style	Training Loss	Validation Loss	Test Loss
Break (3D)	−2.3893	−2.3462	−2.0083
Lock (3D), sequence A	−2.0680	−1.3123	−1.4806
Lock (3D), sequence B	−2.0514	−1.8568	−1.4469
Street Jazz (2D)	−3.1550	−3.1635	−1.3735
Ballet Jazz (2D)	−1.7910	−1.4471	0.4335

and more genres in the table would be needed for such statements, which is part of our future plans. All we can say is that in these runs, held-out NLL fell short of training NLL, substantially so in some runs, and in one case the test NLL rose above zero. This implies that even where a model is trained well, unseen portions of dances may contain transitions outside the learned 30-frame window that the model has never seen and finds difficult to predict. Whether this is a problem inherent to the style, the modality, the length of the windows, or just the luck of the splits, we leave for future work.

Qualitative Visual Analysis

In our current setting, we do not quantify the properties discussed below and defer a deeper investigation to future iterations. As an absolute pose baseline was not trained in the same setting, we cannot attribute any of the following to the use of deltas, and we report them as properties of the system as a whole. Upon visual inspection of the results, we observe that the rollout motion sequences remain largely continuous and do not degenerate to a mean pose. Furthermore, we observe that the rollouts retain local trajectory information in fast extremity motion, particularly of the wrists and ankles, which are usually difficult to model because they move quickly and vary widely across styles. This observation is consistent with the design argument for the delta target, but we cannot make any strong claims to this effect at this time. As for likelihood, we qualitatively observed that models with lower held-out NLL produce smoother motions, but we do not have comparisons for the same style to make any claims in this direction.

We also performed some informal checks to investigate the stochasticity of the model. For a fixed seed, varying the noise vector leads to visibly different motions, which is the intended benefit of the probabilistic output. For diversity, we would like to display side-by-side samples and a quantitative diversity metric, but we defer this to the next iteration.

Figures 5 and 6 show that the skeleton stays connected during generation: the limb-length correction prevents the most visible joint-separation failures, and the head stays anchored relative to the body. The same figures expose the cost of a hard geometric constraint, since very large or stylized extensions

are visibly softened. The output therefore functions as a structurally readable draft, consistent with the assistive role defined in the Objectives, and it stops short of a finished phrase.

Discussion

Regarding the claim that delta targets reduce freezing, the empirical evidence presented here in Table 3 does not isolate this factor from others, so while the observations remain valid and consistent with the alternative hypothesis, the claim that the target representation is the determining factor is unproven without baselines trained identically but with absolute-pose targets. The observed smoothness could come from the Transformer architecture, the Gaussian likelihood, the sampling temperature, the skeletal correction, or the dataset, and the NLL values are interpretable only relative to one another. We therefore plan to conduct a more systematic ablation and comparison study, comparing freezing behavior in a variety of conditions, with deltas or absolute poses, with NLL or MSE, with or without post-hoc correction, and with mean values or sampled sequences. All with identical splits and rollouts for comparability, we will measure freezing with metrics like frame-to-frame velocity magnitude, decay of mean acceleration, variance of pose over the rollout, and the fraction of frames below a motion threshold.

The model also exhibits significant overfitting, as test NLL is worse than training NLL in all runs, and severely so in the weakest cases. This overfitting is partially due to the limited amount of data per model, only 7 to 20 s of motion against about 500k parameters, so the gaps are consistent with the model largely memorizing its training windows, but it also indicates that the model has not learned to interpolate well within the range of a single dance sequence. The delta formulation does not by itself guarantee that a short fixed window captures the longer-range structure of a genre, and with five runs across four genres we cannot say which factor drives the degradation, so repeated runs per style are needed before any style-level claim.

In comparison to the full dance generation models FACT⁸, Bailando¹⁴, and EDGE¹⁵, our goal was to produce results on

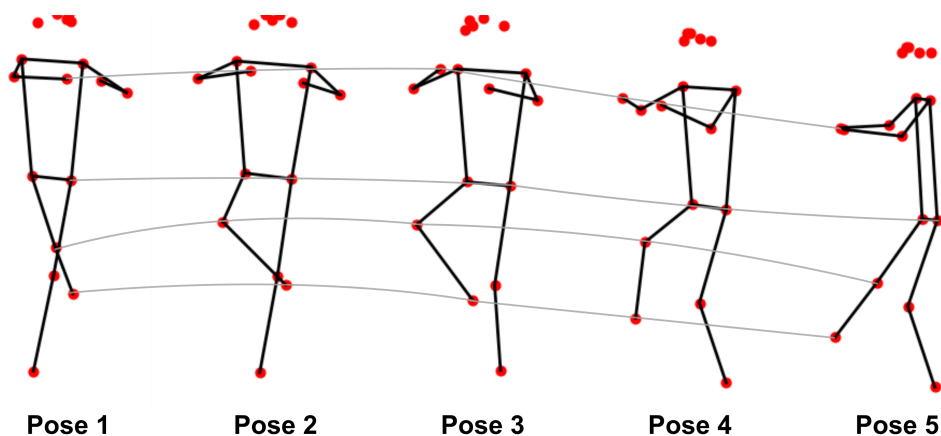


Figure. 5 Sequential visualization of generated 2D Ballet Jazz keypoint transitions. The gray guide lines show how corresponding joints move across the sampled poses; note that limb lengths stay stable from pose to pose while the arms and legs continue their trajectories.

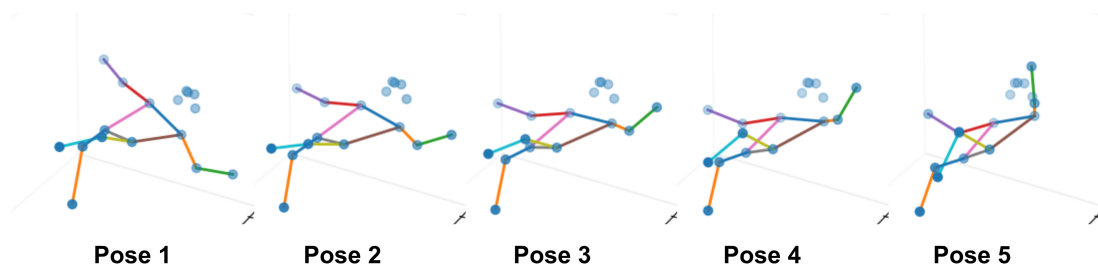


Figure. 6 Sequential visualization of generated 3D Break keypoint transitions, illustrating that connected skeletal structure is maintained during a dynamic, low-body-position movement; the head remains anchored to the shoulder line throughout.

a narrower task, namely dance continuation without the influence of music. While FACT and Bailando provide strong baselines for the task of music-conditioned dance generation with different pose representations, such as SMPL rotations or quantized code sequences, their large model sizes and compute budgets make them difficult to compare to our approach in a meaningful way. The EDGE system also generates full sequences of choreography to a given music input, and does include a large user study with dancers and distributional metrics for measuring properties of motion. Because the pose representations and metrics differ across these works as well, our per-coordinate Gaussian NLL over deltas is not directly comparable to the metrics previously reported. We view our model as a lightweight, likelihood-based counterpart to these systems, useful for studying the continuation task in isolation, and any claim beyond that awaits evaluation on shared metrics.

The pose correction is at once the main safeguard and the main limitation of the system. While it keeps the skeleton in

a readable configuration, there is a trade-off between recovery of the correct skeleton and recovery of the correct style or character of the movement. The correction process measures the head position and the lengths of limbs from the seed, which keeps the skeleton from separating at joints, but it softens extreme extensions. It is not clear where the line is between a physically impossible pose and a stylistic one, and our evidence for this trade-off is qualitative; quantifying it, for example by measuring how much the correction shortens the highly expressive wrist trajectories in Waacking, is future work. Because the post-hoc correction is applied after generation, it cannot resolve this ambiguity, so in future work we will try modeling this as an anatomical prior and applying the correction as a soft penalty during training, to explore how much of this trade-off can be learned.

Lastly, the current system is quite limited in scope. The model is very small and efficient, a 3-layer encoder run once per frame, and this is a strength of this work because it allows focused research on the continuation task. However, the

model does not condition on music, does not model foot interaction or compositional intent, and does not select among the continuations it produces. Assessment of whether the delta model produces a meaningful change in dynamics is limited to researcher inspection, so its usefulness to working choreographers remains unmeasured. Its demonstrated contribution is narrower: from a single seed phrase it returns structurally readable candidate continuations at low computational cost, and whether that shortens real drafting time is the question a user evaluation must answer.

In the spirit of EDGE's distribution-based metrics and user study, we intend to conduct user studies with dancers and choreographers, even at small scale, to evaluate the effectiveness of the system. We also plan to expand the model architecture to more expressive representations and controls, such as parametric 3D pose with SMPL²⁹ instead of keypoints, learning to account for feet-ground interaction, and conditioning the model on music and on controlled style prompts. For evaluation, we intend to calculate jerk, the rate of change of acceleration, and other metrics not directly learned in the model or included in the NLL, such as bone length error, foot sliding, head position error, and diversity of continuations with multiple samples. These will all be compared against baseline models.

Conclusion

In the end, we are modest in our claims. We have shown a seed-conditioned probabilistic Transformer model that can extend short 2D or 3D skeletal dance phrases into longer sequences. These continuations show continued motion, retain some of the local rhythm and spatial structure of the seed, and generally maintain readable, connected skeletons. As stated in the Introduction, our model achieved some but not all of its stated goals. We succeeded at developing a unified framework for 2D and 3D, applying our skeletal correction method during generation, and using a mix of likelihood and visual analyses for model evaluation. On the other hand, we did not test the hypothesis that our delta targets helped reduce freezing, as we did not train an absolute-pose baseline.

In the Discussion section, we outlined directions for further improvement. We recommend conducting matched ablations, using metrics besides teacher-forced likelihood, including evaluation at the sequence rollout level, per-style metrics, and other metrics relevant to dance, and using sequence-level splits, including dancer-held-out splits. We also recommend gathering dancer feedback in an organized way. Future work should include all of these suggestions so as to substantiate claims about what does and does not matter in this modeling task. In our own work, held-out likelihood differed across runs, and training likelihood was always higher. We found that the post-hoc skeletal correction method did prevent ma-

jor anatomical failures in the generated results, although at the cost of some of the more stylized movements. We note that these findings come with the limitations we discussed earlier, since we did not compare against a baseline or run a user study, our only quantitative metric is teacher-forced likelihood, and our splits are at the window level.

We believe it is interesting to consider how to expand the set of possibilities considered in a dance piece. This is always, of course, a creative project with the choreographer as the final judge, but we are hoping that we can build more tools to help. In this work, we have found it interesting and helpful to consider a computationally interactive version of this task, and to consider the constrained case of dance continuation separately from larger efforts to model or simulate choreography given music.

References

- 1 F. Sagasti. Information technology and the arts: the evolution of computer choreography during the last half century. *Dance Chronicle*. Vol. 42, pg. 1–52, 2019, <https://doi.org/10.1080/01472526.2019.1575661>.
- 2 K. Fragkiadaki, S. Levine, P. Felsen, J. Malik. Recurrent network models for human dynamics. *Proceedings of the IEEE International Conference on Computer Vision*. pg. 4346–4354, 2015, <https://arxiv.org/abs/1508.00271>.
- 3 L. Crnkovic-Friis, L. Crnkovic-Friis. Generative choreography using deep learning. *Proceedings of the Seventh International Conference on Computational Creativity*. pg. 272–277, 2016, <https://arxiv.org/abs/1605.06921>.
- 4 R. Huang, H. Hu, W. Wu, K. Sawada, M. Zhang, D. Jiang. Dance revolution: long-term dance generation with music via curriculum learning. *Proceedings of the International Conference on Learning Representations*. 2021, <https://arxiv.org/abs/2006.06119>.
- 5 J. Martinez, M. J. Black, J. Romero. On human motion prediction using recurrent neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pg. 2891–2900, 2017, <https://arxiv.org/abs/1705.02445>.
- 6 D. Pavllo, D. Grangier, M. Auli. QuaterNet: a quaternion-based recurrent model for human motion. *Proceedings of the British Machine Vision Conference*. 2018, <https://arxiv.org/abs/1805.06485>.
- 7 A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*. Vol. 30, pg. 5998–6008, 2017, <https://doi.org/10.48550/arXiv.1706.03762>.
- 8 R. Li, S. Yang, D. A. Ross, A. Kanazawa. AI choreographer: music conditioned 3d dance generation with AIST++. *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pg. 13401–13412, 2021, <https://arxiv.org/abs/2101.08779>.
- 9 S. Tsuchida, S. Fukayama, M. Hamasaki, M. Goto. AIST dance video database: multi-genre, multi-dancer, and multi-camera database for dance information processing. *Proceedings of the 20th International Society for Music Information Retrieval Conference*. pg. 501–510, 2019, <https://archives.ismir.net/ismir2019/paper/000060.pdf>.
- 10 E. Aksan, M. Kaufmann, P. Cao, O. Hilliges. A spatio-temporal transformer for 3d human motion prediction. *Proceedings of the 2021 International Conference on 3D Vision (3DV)*. pg. 565–574, 2021, <https://doi.org/10.1109/3DV53792.2021.00066>.
- 11 E. V. Mascaró, S. Ma, H. Ahn, D. Lee. Robust human motion forecasting using transformer-based model. *2022 IEEE/RSJ International Conference*

- on Intelligent Robots and Systems (IROS). pg. 10674–10680, 2022, <https://doi.org/10.1109/IROS47612.2022.9981877>.
- 12 H. Liu, X. Han, C. Jin, L. Qian, H. Wei, Z. Lin, F. Wang, H. Dong, Y. Song, J. Xu, Q. Chen. Human motionformer: transferring human motions with vision transformers. Proceedings of the International Conference on Learning Representations. 2023, <https://arxiv.org/abs/2302.11306>.
 - 13 H.-Y. Lee, X. Yang, M.-Y. Liu, T.-C. Wang, Y.-D. Lu, M.-H. Yang, J. Kautz. Dancing to music. Advances in Neural Information Processing Systems. Vol. 32, 2019, <https://arxiv.org/abs/1911.02001>.
 - 14 L. Siyao, W. Yu, T. Gu, C. Lin, Q. Wang, C. Qian, C. C. Loy, Z. Liu. Bailando: 3d dance generation by actor-critic GPT with choreographic memory. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pg. 11050–11059, 2022, <https://arxiv.org/abs/2203.13055>.
 - 15 J. Tseng, R. Castellon, C. K. Liu. EDGE: editable dance generation from music. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pg. 448–458, 2023, <https://arxiv.org/abs/2211.10658>.
 - 16 D. P. Kingma, M. Welling. Auto-encoding variational Bayes. 2nd International Conference on Learning Representations. 2014, <https://doi.org/10.48550/arXiv.1312.6114>.
 - 17 M. Petrovich, M. J. Black, G. Varol. Action-conditioned 3d human motion synthesis with transformer VAE. Proceedings of the IEEE/CVF International Conference on Computer Vision. pg. 10985–10995, 2021, <https://doi.org/10.1109/ICCV48922.2021.01080>.
 - 18 I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. Generative adversarial nets. Advances in Neural Information Processing Systems. Vol. 27, pg. 2672–2680, 2014, <https://arxiv.org/abs/1406.2661>.
 - 19 J. Ho, A. Jain, P. Abbeel. Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems. Vol. 33, pg. 6840–6851, 2020, <https://doi.org/10.48550/arXiv.2006.11239>.
 - 20 G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-Or, A. H. Bermano. Human motion diffusion model. Proceedings of the International Conference on Learning Representations. 2023, <https://arxiv.org/abs/2209.14916>.
 - 21 G. E. Henter, S. Alexanderson, J. Beskow. MoGlow: probabilistic and controllable motion synthesis using normalising flows. ACM Transactions on Graphics. Vol. 39, No. 6, pg. 1–14, 2020, <https://doi.org/10.1145/3414685.3417836>.
 - 22 G. Valle-Perez, G. E. Henter, J. Beskow, A. Holzapfel, P.-Y. Oudeyer, S. Alexanderson. Transflower: probabilistic autoregressive dance generation with multimodal attention. ACM Transactions on Graphics. Vol. 40, pg. 1–14, 2021, <https://arxiv.org/abs/2106.13871>.
 - 23 Y. Yuan, K. Kitani. DLow: diversifying latent flows for diverse human motion prediction. Computer Vision - ECCV 2020. pg. 346–364, 2020, <https://arxiv.org/abs/2003.08386>.
 - 24 T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick. Microsoft COCO: common objects in context. Computer Vision - ECCV 2014. Vol. 8693, pg. 740–755, 2014, https://doi.org/10.1007/978-3-319-10602-1_48.
 - 25 M. R. Ronchi, P. Perona. Benchmarking and error diagnosis in multi-instance pose estimation. Proceedings of the IEEE International Conference on Computer Vision. pg. 369–378, 2017, <https://doi.org/10.1109/ICCV.2017.48>.
 - 26 OpenMMLab. 2d body keypoint datasets. MMPose documentation. https://mmpose.readthedocs.io/en/latest/dataset_zoo/2d_body_keypoint.html, 2021.
 - 27 I. Loshchilov, F. Hutter. Decoupled weight decay regularization. Proceedings of the 7th International Conference on Learning Representations. 2019, <https://doi.org/10.48550/arXiv.1711.05101>.
 - 28 J. D. Hunter. Matplotlib: a 2d graphics environment. Computing in Science Engineering. Vol. 9, pg. 90–95, 2007, <https://doi.org/10.1109/MCSE.2007.55>.
 - 29 M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, M. J. Black. SMPL: a skinned multi-person linear model. ACM Transactions on Graphics. Vol. 34, No. 6, pg. 1–16, 2015, <https://doi.org/10.1145/2816795.2818013>.

Appendix: Acronyms

Table 4 Acronyms used in this paper, in order of first appearance.

Acronym	Meaning
NLL	Negative Log-Likelihood, the training and evaluation objective
AIST++	The dance motion dataset built from the AIST Dance Video Database
LSTM	Long Short-Term Memory, a gated recurrent neural network
GRU	Gated Recurrent Unit, a related gated recurrent network
FACT	Full-Attention Cross-modal Transformer, the model introduced with AIST++
GPT	Generative Pre-trained Transformer
VAE	Variational Autoencoder
GAN	Generative Adversarial Network
DDPM	Denoising Diffusion Probabilistic Model
COCO	Common Objects in Context, the source of the 17-keypoint format
SMPL	Skinned Multi-Person Linear model, a parametric 3D body model
MSE	Mean Squared Error
ReLU	Rectified Linear Unit activation function
AdamW	Adam optimizer with decoupled weight decay