Invisible Strings: Revealing Latent Dancer-to-Dancer Interactions with Graph Neural Networks

Luis Vitor Zerkowski[1], Zixuan Wang[2], Ilya Vidrin[3], Mariel Pettee[4]

[1] University of Amsterdam luisvz@gmail.com
[2] Georgia Institute of Technology zxwang9811@gmail.com
[3] Northeastern University i.vidrin@northeastern.edu
[4] Lawrence Berkeley National Lab mpettee@lbl.gov

Abstract

Dancing in a duet often requires a heightened attunement to one's partner: their orientation in space, their momentum, and the forces they exert on you. Dance artists who work in partnered settings might have a strong embodied understanding in the moment of how their movements relate to their partner's, but typical documentation of dance fails to capture these varied and subtle relationships. Working closely with dance artists interested in deepening their understanding of partnering, we leverage Graph Neural Networks (GNNs) to highlight and interpret the intricate connections shared by two dancers. Using a video-to-3D-pose extraction pipeline, we extract 3D movements from curated videos of contemporary dance duets, apply a dedicated pre-processing to improve the reconstruction, and train a GNN to predict weighted connections between the dancers. By visualizing and interpreting the predicted relationships between the two movers, we demonstrate the potential for graph-based methods to construct alternate models of the collaborative dynamics of duets. Finally, we offer some example strategies for how to use these insights to inform a generative and co-creative studio practice.

Introduction

In many cultures, dance is deeply informed by interpersonal connection. The prevalence of social dances worldwide illustrates how dance can even, in certain contexts, be seen as an inherently interactive art form. The collaborative dimensions of dance, however, are less concrete than body positions or movements, and for this reason they can be more difficult to describe or study. Despite their invisible nature, these inter-dancer connections can provide an essential lens through which we can create and understand choreography involving more than one person.

Moreover, in a culture that increasingly values digital renderings of ideas, there is significant potential for dancers to re-imagine how digital forms of dance can enrich, rather than flatten, their understanding of the medium. We present an artist-driven methodology that uses neural networks to investigate the invisible connections between pairs of dancers. While machine learning techniques have been applied to solo dance performances, the intricate realm of partnering, which involves constant negotiation of weight shifts, synchronized gestures, and mutual influence, has remained relatively unexplored. We address this gap by using AI to capture subtle interactions between two dancers.[1]

We employ a pose extraction system on video recordings from the Partnering Lab (Vidrin 2025), translating 2D footage into 3D body poses in collaboration with the original dancers. This conversion enables us to track the trajectories of body joints and collect detailed data on each dancer's movement, later used to understand their collective patterns.

We use Graph Neural Networks (GNNs) to propagate information through the dancers' bodies, modeled as a set of nodes connected by edges. We also use Recurrent Neural Networks (RNNs) to handle the temporal nature of dance, allowing us to better capture the evolution of movements frame by frame. Because labeling how dancers influence each other is highly subjective, we adopted a self-supervised learning approach, optimizing for sequence reconstructions and focusing on the discovery of novel or unanticipated patterns rather than enforcing specific interactions.

In this proof-of-concept work, we present results using only a subset of points from each dancer instead of the full dimensionality of the input data. We also validate our methodology using a particle dataset with a known latent interaction graph. We find that the GNN model is able to identify interesting patterns in between pairs of dancers that align with the dancers' own intuitions. Looking ahead, we suggest that this methodology has the potential to complement embodied dance research strategies by highlighting surprising and invisible connections, reflecting and refining the dancers' understandings of their own creative relationships. In doing so, we not only extend our understanding of how dancers and technologists can collaborate, but also offer new ways for dancers and choreographers to perceive and shape their own practice.

Related Work

Graph Structure Learning Given our research interests in modeling the relationships between dancers' movements, the field of graph structure learning is highly relevant. We can relate our model to it by treating each body joint as

[1] https://github.com/humanai-foundation/ChoreoAI/tree/main/ChoreoAI_Duet_ChorAIgraphy_Luis_Zerkowski

a node and thinking of edges as capturing inter-joint (or inter-dancer) interactions. Graph Convolutional Networks (GCNs) (Kipf and Welling 2017) extend convolutional operations to graph-structured data, allowing message passing between connected nodes. For time series data like dance, Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber 1997), can capture temporal dependencies by maintaining hidden states over frame sequences. Variants such as Graph Recurrent Neural Networks (GRNNs) (Ruiz, Gama, and Ribeiro 2020; Li et al. 2017a) combine these ideas, propagating information through both graph edges and time.

Variational Autoencoders on Graphs To learn the underlying structure of these dancer-to-dancer interactions, we employ methods inspired by Variational Autoencoders (VAEs) (Kingma and Welling 2022). VAEs have proven effective in unsupervised representation learning, enabling models to discover latent factors from complex input data. Building on this foundation, Neural Relational Inference (NRI) (Kipf et al. 2018) introduces a framework specifically designed to infer interaction graphs within systems of multiple nodes. NRI parameterizes the probability of different edge types - each one representing distinct forms of interaction - and learns these latent relationships.

AI and Dance Researchers have generated 2D and 3D dance movements using a variety of machine learning techniques including RNNs (Graves 2013; McCormick et al. 2015; Crnkovic-Friis and Crnkovic-Friis 2016; Alemi, Françoise, and Pasquier 2017; Li et al. 2017b; James 2018; Marković and Malešević 2018), clustering strategies like Kernel Principal Component Analysis (KPCA) (Berman and James 2015; Schölkopf, Smola, and Müller 1998), VAEs (Pettee et al. 2019; Papillon, Pettee, and Miolane 2022), GNNs (Pettee et al. 2020), and Transformers conditioned on musical inputs (Li et al. 2021b; 2020).

The NRI framework was previously applied to dance data by Pettee et al., but this work only considered the movements of a solo dancer. By adapting NRI to choreographic duets and enhancing its capacity of temporal understanding with GRNNs, we aim to automatically detect how one dancer's movements influence the other, ultimately revealing what the model identifies as subtle partnering dynamics that might otherwise go unnoticed.

Dataset Preparation

This project required transforming raw video footage of dance duets into structured pose data while addressing challenges such as multi-person tracking, occlusion, and movement complexity. We detail the dataset curation, pose extraction, and post-processing steps to ensure high-quality data for analysis and model training.

Data Collection

As an artist-centric effort, our project prioritized the needs and agency of our creative collaborators. We worked closely with Dr. Ilya Vidrin, leader of the Partnering Lab (Vidrin 2025), to develop the scope of the project and curate the data to suit our research goals. We collected video footage of movement data from duets of dancers associated with the Partnering Lab, totalizing four videos with an aggregate duration of 41 minutes and an average duration of 10 minutes and 15 seconds. Each video was filmed with a similar single-camera setup that started with each dancer facing one another on opposite sides of the screen, each grasping the other's right hand. The dancers featured in the filmed videos opted into the project and consented to our team using the data to train models designed to analyze duets for this project.

The movement styles of the dancers reflect the practices of the Partnering Lab, which "focuses on the internal, kinesthetic experience of movement that is difficult to perceive outside of experience" (Vidrin 2025). Targeting micromovements and exploring partnering as an embodiment of ethical concepts including trust and care, the dancers in these duets are highly attuned to one another, often moving slowly and deliberately, exploring mutual tensions and weight with active curiosity. Notably, the Partnering Lab team leader also regularly attended technical collaboration meetings and provided essential feedback that influenced the analysis design throughout the entire project duration.

Pose Extraction Pipeline

2D Pose Extraction We initially experimented with 2D pose estimation models from *AlphaPose* (MVIG-SJTU 2025), but found them insufficient for capturing spatial relationships between dancers. Also, despite the recent advances in human body pose extraction from images and multi-person tracking (Cao et al. 2017; Wei et al. 2016; Simon et al. 2017; Cao et al. 2019; Fang et al. 2017; Li et al. 2019), issues such as frame drops, identity swaps, and joint jitter were frequent, requiring extensive post-processing. An extraction example is shown in Figure 1.

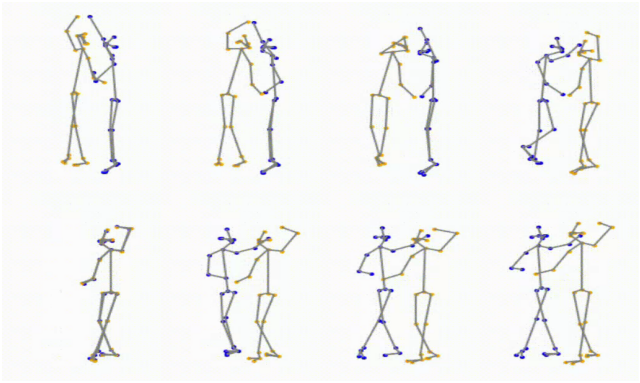


Figure 1: A raw 2D pose sequence from the Halpe pretrained model (26 keypoints) that shows vertex noise and even a missing dancer at one timestep.

3D Pose Extraction Given the limitations of 2D methods, we adopted 3D pose estimation and 3D mesh reconstruction

pipelines (Li et al. 2021a; Kocabas, Athanasiou, and Black 2020) to preserve depth information crucial for analyzing dancer interactions. We evaluated two models:

- *VIBE* (Kocabas 2025): A prominent 3D pose and shape estimation model that outputs both joint positions and mesh reconstructions.
- *HybrIK* (integrated with *AlphaPose* (MVIG-SJTU 2025)): A hybrid analytical-neural approach that refines pose estimation by combining classical kinematics with learning-based corrections.

In Figure 2, we show an image comparison to visualize the resulting 3D joints and meshes.

(a) Mesh reconstruction from 3D pose coming from *VIBE*.

(b) Mesh reconstruction from 3D pose coming from *HybrIK*.

Figure 2: Comparison of 3D pose extractions: *HybrIK* (bottom) outperforms *VIBE* (top) in both simple (stationary) or complex (dynamic) movements.

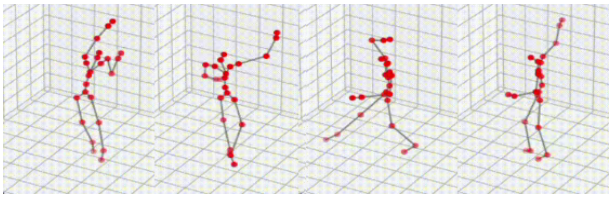
After testing, *HybrIK* (via *AlphaPose*) was selected for its higher pose accuracy, better multi-person tracking, and improved pose consistency. While minor noise and missing frames persisted, *HybrIK*'s outputs were the most reliable for partnered dance sequences.

Data Cleaning

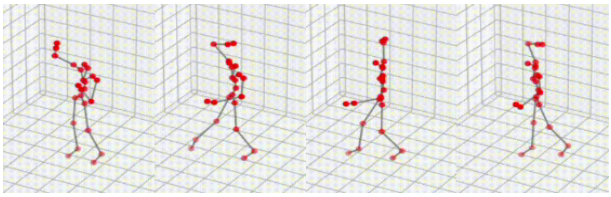
Despite selecting the best-performing model, raw 3D pose data still contained noise and inconsistencies. To ensure stable movement trajectories, we applied several post-processing steps:

- **Handling Missing Frames:** If a frame had no detected poses, we duplicated the poses from the previous frame. Since our videos run at around 30 FPS, this small gap-filling strategy did not produce abrupt motion artifacts.
- **Frames with Single-Person Detections:** When only one dancer was detected, we compared the sum of Euclidean distances between corresponding joints for the identified person and the two people in the previous frame. We then added the person from the previous frame with the greater distance to the current frame (assuming this was the non-captured person).
- **Frames with More Than Two Detections:** Some frames falsely identified more than two individuals. Since our videos contained exactly two dancers, we kept only the two highest-confidence detections.
- **Index Consistency:** If dancers swapped IDs across frames, we scanned correct frames around the confusion frame and used the aforementioned sum of Euclidean distances between corresponding joints to correct the inversions.
- **Vertex Jitter:** Even successful detections contained local noise, causing "shaky" joints. We applied a 3D Discrete Cosine Transform (DCT) (Ahmed, Natarajan, and Rao 1974) low-pass filter at a 25% threshold to smooth out high-frequency jitter.

The impact of these corrections can be seen in Figure 3 by comparing the minimally-processed (i.e. handling missing frames and missing/additional people) output with our fully-processed result. The final output is cleaner, more stable, and better preserves each dancer's bodily trajectory, which is vital for dance partnering research.



(a) Pose extraction with minimal processing.



(b) Pose extraction with full processing pipeline.

Figure 3: Comparison of 3D pose extractions for a dancer: minimal processing (top) and full pipeline (bottom). While both depict the same sequence, the bottom image captures the movement far more clearly, with smoother, more continuous transitions between frames. In contrast, the top image appears fragmented, with disconnected poses that obscure the flow of motion.

3D Graph Construction

To model dancer interactions, we represented each dancer as a graph of 29 joints (nodes), a skeleton obtained from

HybrIK (Li 2025)[2], and fully connected all joints between dancers to form a dense bipartite graph. The learning objective is to classify or weight these inter-dancer edges, indicating how crucial each connection is for movement correlation.

The data is then prepared for model training by creating batches with *PyTorch* (Paszke et al. 2019) tensors. The tensors are structured with dimensions representing the total number of sequences, the sequence length, the number of joints from both dancers, and 3D coordinates + 3D velocity estimates. Finally, a 85%-15% training-validation split is created to allow for proper model hyperparameter tuning. To improve model generalization given our limited dataset, the training pipeline incorporates a data augmentation step that involves rotating batches of data. Each batch is rotated along the $\hat{z}$-axis by a randomly-selected angle $\theta \in [0, 2\pi]$ while maintaining the original $\hat{x}$ and $\hat{y}$-axis orientations for physical consistency. This approach helps prevent the model from overfitting to the dancers' absolute positions.

Due to the high complexity of the problem, both in the number of moving nodes and the number of edges in the graph, random joint sampling was implemented to reduce the scale of the problem. Only subsets of 6 to 10 joints (3 to 5 from each dancer) are used in each training run, and only inter-dancer edges among those sampled joints are considered. This approach helps the network converge more reliably while demonstrating proof-of-concept for larger graphs.

Graph Neural Network Model

Understanding dancer interactions requires modeling relationships rather than individual trajectories. We extend Neural Relational Inference (NRI) (Kipf et al. 2018) to treat joints as graph nodes and infer edges representing inter-dancer connections. This section details the model's formulation and architecture.

Neural Relational Inference Variant

Our model builds on NRI (Kipf et al. 2018), an extension of Variational Autoencoders (VAEs) (Kingma and Welling 2022), originally designed to infer latent interaction graphs from particle motion. By analyzing position and velocity, NRI estimates which particles influence others without predefined relationships.

This approach is particularly relevant for duet dance analysis. Just as NRI infers particle interactions without a known ground-truth graph, we also lack a predefined interaction graph for dance partnering. The relationships between dancers shift dynamically, and there's no ground-truth connection between dancers. Instead, we employ self-supervised learning, optimizing for sequence reconstruction rather than enforcing predefined structures.

Our model consists of an encoder and a decoder, both designed to iteratively transform node representations into edge representations and vice versa. The encoder generates edge logits, which are used to construct the latent space. Since edge indices are later sampled for the decoder, transitioning between representations is a crucial step in the process. Our model implementation includes a few important modifications from the core NRI structure (Kipf et al. 2018):

- **Graph Convolutional Network (GCN):** Some linear layers are replaced with GCN layers to leverage the graph structure, improving the model's ability to capture relationships between joints. This change focuses on a subset of edges connecting both dancers rather than studying all joint relationships - as in the original implementation. Additionally, GCNs provide local feature aggregation and parameter sharing, important inductive biases for the context.
- **Graph Recurrent Neural Network (GRNN) Decoder:** To make better use of sequential information and potentially achieve a more suitable final embedding for predicting (or reconstructing) the next frame, one possible approach is to use a recurrent network. It is important to note that this is more of an update to a version of the NRI model rather than a completely new idea. The authors had already explored a recurrent decoder, with the main differences being the previously mentioned modifications in layers and the use of GRU nodes instead of LSTM nodes (introduced in the next item).
- **Custom GCN-LSTM Cells:** To utilize the recurrent structure crucial for sequence processing while maintaining graph information and GNN architecture, the classic LSTM cell has been reimplemented with GCN nodes. In the final version of our architecture, only the decoder incorporates the recurrent component, which generates a final sequence embedding that the model uses to reconstruct the next frame.

By incorporating these modifications, the model maintains the core principles of the original NRI while theoretically enhancing its ability to generalize and adapt to the dynamics of "connected" particles moving. We now describe the architecture (shown in figure 4) more precisely:

Encoder

- **Node-to-Edge Transformation:** Dancer joints are passed through a GCN layer with 64 latent dimensions and then converted to edge representations, which doubles this representation.
- **Linear Layer + Batch Normalization + Dropout:** A linear layer refines edge embeddings, receiving input of dimension 128 dimensions and generating an output of 64 dimensions. Optionally we use batch normalization and dropout with 10% probability for regularization.
- **Edge-to-Node Transformation:** The edge representations are converted back into nodes, and another GCN layer with same dimensionality as before is applied.
- **Second Node-to-Edge Transformation:** The nodes are then transformed back into edges, followed by another Linear layer from 192 dimensions (this time we have

[2]More specifically, from this issue `https://github.com/jeffffffli/HybrIK/issues/140` and expanded through this code `https://github.com/jeffffffli/HybrIK/blob/main/hybrik/datasets/h36m_smpl.py`.

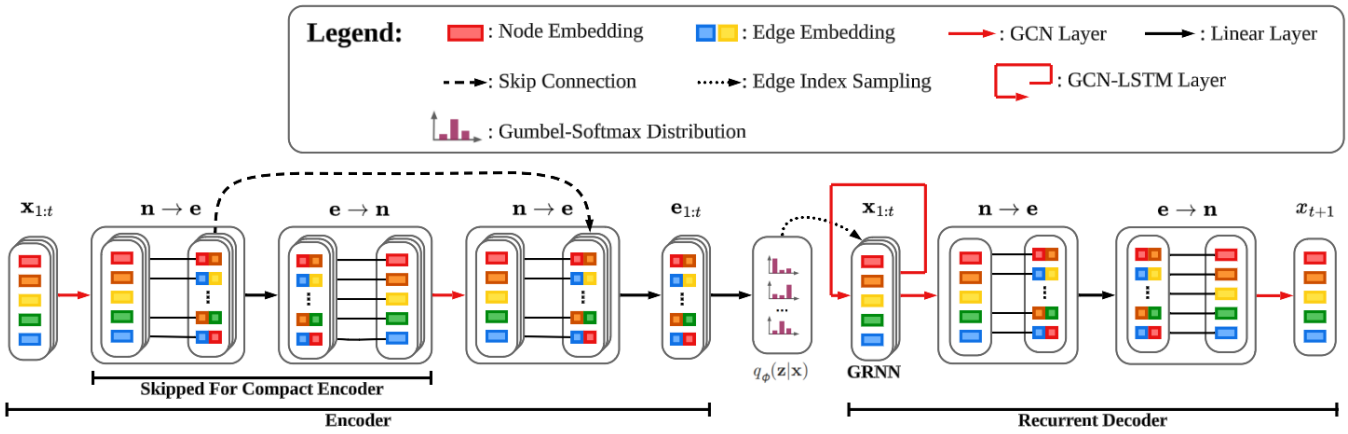


Figure 4: Schematic of the final model architecture, including the GCN nodes and the GRNN adapatation, inspired by the one found in the original NRI paper (Kipf et al. 2018) (Figure 3, page 3). The "compact encoder" variation - marked here as "Skipped for Compact Encoder" - was introduced to test a simplified version of the model, motivated by challenges with data. This version includes only a single node-to-edge transition in the encoder. However, as shown later in Table 1, the compact encoder did not outperform the full model and, in some cases, performed slightly worse. For this reason, it was only used for comparative experiments and was not adopted in the final architecture.

3×64=192 because we also use a skip connection coming from the first Node-to-Edge transformation) to 64 dimensions.

- **Logits Computation:** A final linear layer outputs logits for each possible edge type. For this layer, we mainly tested with binary edges ("existing", "non-existing") or 3 edge types ("no connection," "weak" and "strong"), but we also added the possibility of 4 or even 5 edge types. Each one of these options comes with a prior probability distribution that guides the latent distribution, making the later sampling process gravitate towards a certain allocation of each edge type.

Decoder

- **Edge Index Sampling:** The decoder begins by hard sampling edge indices using a Gumbel-Softmax distribution (Maddison, Mnih, and Teh 2017; Jang, Gu, and Poole 2017), based on the logits generated by the encoder and the prior probabilities of each edge type. While this process results in discrete edges (i.e., edges are either present or absent, without intermediate probabilities), it effectively approximates sampling from a continuous distribution and uses Softmax to enable the reparameterization trick, keeping the entire pipeline fully differentiable.
- **GRNN + Node-to-Edge Transformation:** Once edges have been sampled, the decoder processes the data through a GRNN composed of modified LSTM nodes with GCN layers, followed by a transformation of the final sequence embedding into edge representations with 64 dimensions again.
- **Linear Layer + Batch Normalization + Dropout:** A linear layer refines edge embeddings, receiving input of dimension 128 dimensions and generating an output of 64 dimensions, and optionally using batch normalization and dropout with 10% probability for regularization.
- **Edge-to-Node Transformation:** The edge representations are converted back into nodes, and a GCN layer is applied to predict (or reconstruct) the next frame of the input sequence.

Experiments and Results

We evaluated our model on two datasets: a charged-particle dataset from the original NRI paper, serving as a simpler dataset with a known interaction graph to help benchmark our model performance, and the duet dance dataset, featuring real-world 3D pose sequences with two dancers at a time.

Charged Particle n-Body Simulations

To validate the architecture's viability under more controlled conditions, we used the original charged-particle dataset used in Kipf et al.. We generated 50,000 simulated trajectories of five particles each, with 2D positions and velocities over 49 frames. A random interaction graph dictates repulsive or attractive forces between particles.

We tested a variety of architectures, and the results were promising. Figure 5 presents an example from one of our models, showing that the predicted graph structures and particle trajectories generally aligned with the ground truth. Additional validation results can be found in our open-source codebase.

3D Dance Duets

To assess model performance on the 3D dance data, we use predicted movement fidelity as a proxy for evaluating sampled edges, given that the subjectivity of defining correct inter-dancer connections makes defining true edge labels difficult. Models were trained until convergence, typically

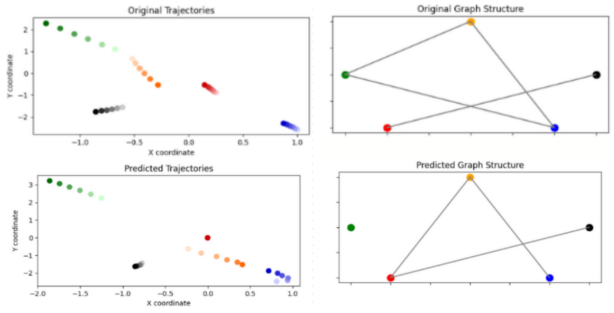


Figure 5: On top, original simulated trajectories and original sampled edges. On bottom, reconstruction and edge prediction results. The model accurately captured the movement and location of three particles (green, black, blue), approximated movement shape for one (orange) despite location inaccuracies, and positioned the last (red) reasonably well but without movement.

for 15 to 20 epochs, and resulted in stable reconstruction loss curves with no signs of overfitting on validation data.

Table 1: **Reconstruction Mean Squared Error** for multiple tasks, input sequence lengths (ℓ), number of edge types (n), and model architecture configurations (compact encoder vs. full architecture as introduced in Figure 4). Bold highlights the best (smallest) reconstruction error.

Task	ℓ	n	Model	MSE
5-Body Sim.	6	3	Compact Enc.	0.70
5-Body Sim.	6	3	Full Arch.	0.69
5-Body Sim.	6	4	Compact Enc.	0.82
5-Body Sim.	12	3	Compact Enc.	0.33
5-Body Sim.	12	3	Full Arch.	**0.32**
6-Joint Dance	8	4	Full Arch.	0.58

KL-divergence loss, however, plateaued early, limiting latent space exploration. While beta coefficient scaling mitigated this effect, short training runs reduced its effectiveness. Longer training stabilized KL behavior, but these experiments were conducted on earlier, smaller models without a Graph Recurrent Neural Network (GRNN), which made extended training less feasible in the final architecture.

For one of our best-performing models, we trained with 6 sampled joints (3 per dancer), with dancer rotation augmenting the dataset 10 times. The full encoder was used with 8-frame input sequences, 64 hidden dimensions, and 4 edge types. It was trained for 20 epochs (18 hours) using mean squared error. Table 1 shows some MSE values for multiple trained models.

Movement Predictions We consider a random subset of 3-5 joints for each of the two dancers for both model training and predicted movement evaluations. The predicted movements are highly sensitive to sampled edges, as information in a graph network spreads through connected nodes. The best-performing models tend to assign multiple connections to a single joint, enabling smoother information flow between dancers. As seen in Figure 6, sampled joints maintain accurate spatial positioning and move coherently with the body. Given that the sampled connections only consist of joints between the dancers, it's notable that the captured interaction can lead to accurate movement predictions for each of the individual dancers.

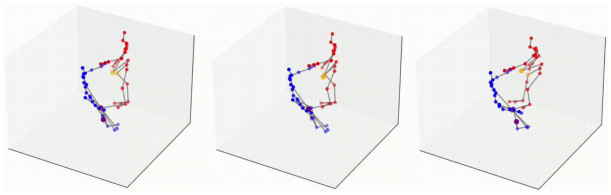


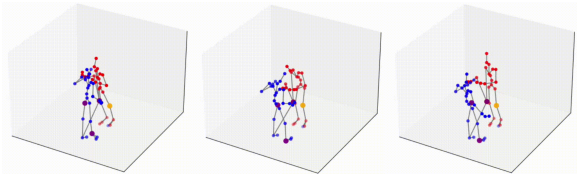
Figure 6: Example of a good reconstruction. Reconstructed sampled joints are color-coded for clarity: purple for the blue dancer and orange for the red dancer.

The reconstructed frames provide valuable insight into how the model interprets connections between particles. By analyzing the learned edge distribution and sampled edges across different examples, it becomes possible to better understand the network's perception of dancer interactions and movement patterns.

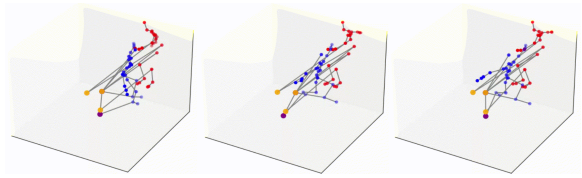
Certain limitations arise due to the nature of the edge sampling and model assumptions. Common sources of reconstruction errors include:

- **Shaking and jitter:** Some particles exhibit noticeable instability, partly due to inherent noise in the original sequence and also because each reconstructed frame is generated independently, without leveraging temporal smoothing since the model predicts only one frame at a time.
- **Limited movement in sampled joints:** In some cases, the sampled edges do not adequately capture movement, resulting in reconstructions in which the joints remain nearly stationary (see e.g. Figure 7, top). This effect is particularly pronounced when the sampled connections fail to establish a strong information pathway between dancers.
- **Challenges with dancer switching:** When dancers cross paths or switch sides, the reconstructed joints sometimes remain near their initial positions rather than following the dancers' actual motion. This suggests that the model has difficulty generalizing to sequences where relative spatial relationships shift significantly.
- **Drift towards the center:** If the dancers move away from the origin, reconstructions tend to stay near the center (Figure 7, bottom), which could be improved using a different form of pre-processing.

Regardless of these challenges, the reconstructions still generally capture meaningful aspects of the dancers' interactions. The observed limitations highlight areas for future refinement, such as improving edge sampling strategies and incorporating temporal consistency in predictions.



(a) Example of edge sampling limiting reconstruction to 3 joints, while the other 3 remain static at the center of the coordinate frame despite being sampled.



(b) Example illustrating the model's reconstruction challenges when dancers are farther from the center, resulting in joints that are either stationary or inaccurately positioned.

Figure 7: Examples of poor reconstructions.

Predicted Edges During inference for edge prediction, edges are sampled differently from the training phase. Instead of hard sampling, which enforces a strict edge selection, we use soft sampling (i.e. sampling with associated probability) to retain only high-confidence connections. This ensures that only the most structurally relevant edges remain in the final reconstruction, reducing noise and improving interpretability.

A key observation is that the number of sampled edges with a confidence above 80% remains consistently low and aligns closely with the prior distribution of edge types (whether two, three, or four types are used). This suggests that the learned latent space effectively captures meaningful movement relationships rather than introducing arbitrary edges. Since the edge priors were designed to reflect a sparse but essential connectivity between dancers, this alignment reinforces the idea that the model is not simply memorizing movement sequences but actively identifying the most influential joints for movement propagation.

The predicted edges exceeding the 80% confidence threshold have interesting several features:

- First, most of these connections tend to have similar confidence levels. This indicates a **low hierarchy among selected edges** (Figure 8). In other words, once an edge is classified as important, it tends to contribute equally to the reconstruction, rather than one edge being clearly dominant. This is particularly interesting from a movement analysis perspective because it suggests that information is distributed across multiple interaction points rather than concentrated in a single connection.
- Additionally, it is common for a single joint to be connected by multiple edges, meaning that **certain key joints act as hubs** in the reconstructed interaction graph (Figure 9). This aligns with prior observations that information propagates more effectively when there are multiple connected paths leading to and from the same dancer (through one joint or within two hops). For instance, a dancer's hand might influence their partner's torso, while simultaneously being connected to their partner's own hand or shoulder, helping to propagate multiple layers of movement dependency.

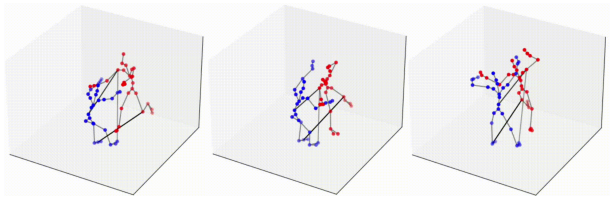
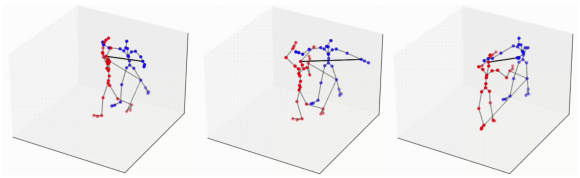
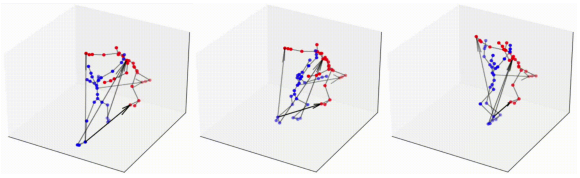


Figure 8: Example of the sampled edge distribution. The black edges represent connections between the dancers, with darker edges indicating higher confidence in their importance for reconstruction. In this typical case, 3 edges were selected for 6 sampled joints, 2 with slightly higher importance, though all exceed 80% confidence.



(a) An undirected example illustrating simple movement for a set of 6 sampled joints.



(b) A directed example showcasing more complex movement for 10 sampled joints. Notably the model captures how the foot motion of one dancer influences multiple parts of the other dancer's body. This aligns with the performed movement, where the blue dancer's dynamic spin guides the red dancer's response.

Figure 9: Examples of multiple edges connected to the same joint.

- Another recurring pattern is that **edges frequently form between joints that are in opposition**, as if connected by an invisible elastic band. These joints appear to be stretching or pulling apart, suggesting that the model is particularly sensitive to moments of tension between dancers (Figure 10). This is especially relevant in the context of duet choreography, where push and pull dynamics are fundamental to many partnering techniques.

  For example, if one dancer extends their arm forward while their partner leans away, the model often identifies a connection between these two points, even if there is no

direct contact. This suggests that the model is not only detecting explicit force transfers (such as corresponding body parts in contact) but also implicit movement dependencies, where one dancer's action influences the other's balance, trajectory, or momentum.

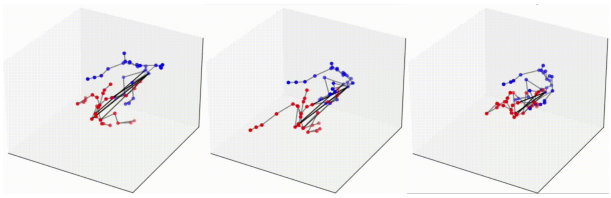


Figure 10: Undirected example of connections within opposition tendencies. It shows multiple connections between the lower torso of both dancers, first leaning in opposite directions and then gravitating toward each other, illustrating the full range of the stretched-string analogy.

This result aligns closely with our dancers' choreographic intuitions and training, in which movement is often defined by how tension and release are negotiated between partners. It also suggests that the model is learning a representation of movement that extends beyond surface-level trajectory following.

Discussion

Creative Implications

This work offers a new, useful lens for dancers to analyze their movements and interactions, revealing patterns and relationships they may not consciously recognize. By mapping inter-dancer connections, it provides insights into habitual tendencies, asymmetries, and unconscious influences in movement. For duets, this tool makes visible the intuitive exchanges between partners — subtle weight shifts, spatial negotiations, and reactive gestures that shape their interaction. Seeing these dynamics mapped out allows dancers to refine their awareness and explore new choreographic possibilities.

This approach reimagines dance as a dynamic graph in which dancers are linked by evolving connections. This perspective could inform interactive performances, projecting real-time movement-based visualizations onto a stage, rendering salient aspects such as tone, frame, timing, and mutual responsiveness. Real-time relational graphs used in performance could project evolving connection structures live, influencing lighting/sound/environment based on detected relational patterns. This would make connection itself an active, evolving part of the performance material — not just a background reality.

This approach can inform the creative process itself (whether improvised or choreographed). Rather than starting with fixed steps or positions, the design of connection dynamics can promote exploration of how energy flows between dancers, how subtle shifts in direction, position, orientation, and point(s) of contact cascade into sequences of movement. Even in highly trained dancers, the relational structures — who is influencing whom, how strongly, at what moment — are felt but not easily seen or measured. Artificial intelligence, particularly relational inference models like this one, can externalize these hidden connection patterns into something visual, trackable, and analyzable over time. Dancers could observe after the fact how influence passed between certain bodies, or that certain points in the body (like hips or hands) carried more relational weight. This would allow dancers to reflect on connection structures in a way that is otherwise invisible — opening possibilities for refinement, experimentation, or re-composition based on those patterns.

Additionally, the system holds potential for dance pedagogy and interdisciplinary research, bridging choreography with biomechanics, for example. Students could receive real-time visualizations of connection strength, delay, or mutuality, transforming abstract instruction into tangible, actionable feedback. Moreover, by analyzing the emergent interaction graphs, dancers could discover non-obvious relationships within their movements, finding novel pathways of influence that could inspire greater awareness, which can subsequently promote greater agency in practice. This system would allow for an emphasis on mutuality rather than hierarchy, honoring connection itself as choreographic material — something that can be composed, taught, and evolved in ways that are otherwise obscured. By exposing hidden layers of movement, this work expands both artistic and analytical possibilities for dancers and technologists.

Future Directions

Despite these promising proof-of-concept results, several improvements are needed to enhance model performance and usability. Key areas for future work include:

- **Data expansion and quality:** Regardless of the use of data augmentation through duet rotation, the small dataset size made training difficult. Also the pipeline used to extract 3D poses has room for improvement. Even in original sequences, the dancers' joints are shaky and often poorly approximated, leading to random (unrealistic) movements. Moreover, normalization of both dancers was removed to preserve relative movement, but it was realized too late that a new layer normalizing their joint movement should have been added - having dancers in different parts of space caused confusion for the models.
- **Architecture exploration:** While many versions of the NRI variant were implemented and tested, the final version is still far from the potential we saw from the strong results achieved by the original version. Furthermore, alternative architectures like transformers (Vaswani et al. 2023) could enhance temporal modeling and better align with modern architectures.
- **Model validation:** Extensive qualitative analysis was conducted through various training sessions, experiments, and even scenario simplifications, but the study did not incorporate a detailed quantitative evaluation beyond learning curve and MSE analysis. Additional metrics and comparative tests with different parameters - such as the number of reconstructed frames - were not much explored and could provide further insights.

- **Processing speed:** Some parts of the final architecture are suboptimal custom implementations. Batches are replaced with sequential operations at several points in the pipeline - node and edge representation conversions, edge sampling for each sequence in a batch, and in the GRNN. As a result, a training cycle with just a few dozen epochs can take an entire day, which is not ideal for scaling.
- **Interaction with dancers:** Since this project sits at the intersection of art and technology, more direct interaction with artists is essential. With a more refined version of the model, it would be ideal to present the tool to the dance community and observe how dancers use the tool in their own partnering studies.

Conclusion

This work presented an investigation into how AI-driven methods can enhance our understanding of partnered dance by focusing on revealing inter-dancer connections. We paired an open-source 3D pose extraction pipeline with a custom pre-processing stage to capture dancers' movements and then trained an NRI-based architecture augmented with GCN modules and RNN structure to reveal important edges between the dancers. Though the model only processed a subset of the full-body kinematics of the dancers, it demonstrated a capacity to uncover interesting dance dynamics such as high-confidence inter-joint dependencies that align with choreographic intuition, the presence of key movement hubs that serve as central points of influence within the duet structure and recurring patterns of tension and release.

While far from comprehensive, our work explores the potential for AI to augment our creative understanding of duet dynamics in choreographic settings. From connections that focus on "push and pull" forces between dancers to more intricate patterns of mutual influence, this project shows the value of graph-based approaches in modeling collaborative creative frameworks.

Author Contributions

Authors one and two jointly developed the pose extraction and data processing pipelines. Author one designed and implemented the machine learning methodology and led the paper manuscript writing. Throughout the process, authors three and four provided technical and artistic supervision.

Acknowledgements

The authors would like to thank the HumanAI program and Google Summer of Code for supporting this work.

References

Ahmed, N.; Natarajan, T.; and Rao, K. 1974. Discrete cosine transform. *IEEE Transactions on Computers* C-23(1):90–93.

Alemi, O.; Françoise, J.; and Pasquier, P. 2017. GrooveNet: Real-time music-driven dance movement generation using artificial neural networks. https://www.researchgate.net/publication/318470738_GrooveNet_Real-Time_Music-Driven_Dance_Movement_Generation_using_Artificial_Neural_Networks.

Berman, A., and James, V. 2015. Kinetic imaginations: Exploring the possibilities of combining AI and dance. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, 2431–2437. AAAI Press.

Cao, Z.; Simon, T.; Wei, S.-E.; and Sheikh, Y. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*.

Cao, Z.; Hidalgo Martinez, G.; Simon, T.; Wei, S.; and Sheikh, Y. A. 2019. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Crnkovic-Friis, L., and Crnkovic-Friis, L. 2016. Generative choreography using deep learning. https://arxiv.org/abs/1605.06921.

Fang, H.-S.; Xie, S.; Tai, Y.-W.; and Lu, C. 2017. RMPE: Regional multi-person pose estimation. In *ICCV*.

Graves, A. 2013. Generating sequences with recurrent neural networks. https://arxiv.org/abs/1308.0850.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

James, A. B. . V. 2018. Learning as performance: Autoencoding and generating dance movements in real time. *Computational Intelligence in Music, Sound, Art and Design* 256–266.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical reparameterization with gumbel-softmax.

Kingma, D. P., and Welling, M. 2022. Auto-encoding variational bayes.

Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks.

Kipf, T.; Fetaya, E.; Wang, K.-C.; Welling, M.; and Zemel, R. 2018. Neural relational inference for interacting systems. *arXiv preprint arXiv:1802.04687*.

Kocabas, M.; Athanasiou, N.; and Black, M. J. 2020. Vibe: Video inference for human body pose and shape estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kocabas, M. 2025. Vibe: Video inference for human body pose and shape estimation. https://github.com/mkocabas/VIBE. Accessed: 2025-02-13.

Li, Y.; Tarlow, D.; Brockschmidt, M.; and Zemel, R. 2017a. Gated graph sequence neural networks.

Li, Z.; Zhou, Y.; Xiao, S.; He, C.; and Li, H. 2017b. Auto-conditioned LSTM network for extended complex human motion synthesis. *CoRR* abs/1707.05363.

Li, J.; Wang, C.; Zhu, H.; Mao, Y.; Fang, H.-S.; and Lu, C. 2019. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10863–10872.

Li, J.; Yin, Y.; Chu, H.; Zhou, Y.; Wang, T.; Fidler, S.; and

Li, H. 2020. Learning to generate diverse dance motions with transformer.

Li, J.; Xu, C.; Chen, Z.; Bian, S.; Yang, L.; and Lu, C. 2021a. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3383–3393.

Li, R.; Yang, S.; Ross, D. A.; and Kanazawa, A. 2021b. Ai choreographer: Music conditioned 3d dance generation with aist++. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* 13381–13392.

Li, J. 2025. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. `https://github.com/jeffffffli/HybrIK`. Accessed: 2025-02-13.

Maddison, C. J.; Mnih, A.; and Teh, Y. W. 2017. The concrete distribution: A continuous relaxation of discrete random variables.

Marković, D., and Malešević, N. 2018. Adaptive interface for mapping body movements to sounds. In Liapis, A.; Romero Cardalda, J. J.; and Ekárt, A., eds., *Computational Intelligence in Music, Sound, Art and Design*, 194–205. Cham: Springer International Publishing.

McCormick, J.; Hutchison, S.; Vincs, K.; and Vincent, J. B. 2015. Emergent behaviour: Learning from an artificially intelligent performing software agent. In *21st International Symposium on Electronic Art*.

MVIG-SJTU. 2025. Alphapose: Real-time and accurate full-body multi-person pose estimation and tracking system. `https://github.com/MVIG-SJTU/AlphaPose`. Accessed: 2025-02-13.

Papillon, M.; Pettee, M.; and Miolane, N. 2022. Pirounet: Creating dance through artist-centric deep learning.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. Pytorch: An imperative style, high-performance deep learning library.

Pettee, M.; Shimmin, C.; Duhaime, D.; and Vidrin, I. 2019. Beyond imitation: Generative and variational choreography via machine learning.

Pettee, M.; Miret, S.; Majumdar, S.; and Nassar, M. 2020. Choreo-Graph: Learning Latent Graph Representations of the Dancing Body. *NeurIPS Workshop on Machine Learning for Creativity and Design*.

Ruiz, L.; Gama, F.; and Ribeiro, A. 2020. Gated graph recurrent neural networks. *IEEE Transactions on Signal Processing* 68:6303–6318.

Schölkopf, B.; Smola, A.; and Müller, K.-R. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 10(5):1299–1319.

Simon, T.; Joo, H.; Matthews, I.; and Sheikh, Y. 2017. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2023. Attention is all you need.

Vidrin, I. 2025. Partnering lab website. `https://www.partneringlab.com/`. Accessed: 2025-02-13.

Wei, S.-E.; Ramakrishna, V.; Kanade, T.; and Sheikh, Y. 2016. Convolutional pose machines. In *CVPR*.