Dance-Language Cross-Modal Embedding: Bridging Movement and Description Through Latent Space Alignment

Ahmed Hassan
Egypt
ahmed.hasan@ejust.edu.eg

***Abstract*—This report presents a comprehensive system for cross-modal embedding between dance movements and natural language descriptions. We introduce a novel pipeline that combines a Variational Autoencoder (VAE) for dance sequence representation, unsupervised clustering for semantic organization, and BERT-based text encoding for language understanding. Through contrastive learning techniques, we establish a unified embedding space where dance movements and textual descriptions become directly comparable and interconvertible. The system demonstrates strong performance in bidirectional mapping between modalities, enabling text-to-motion generation, motion-to-text retrieval, and semantic organization of dance sequences. This work represents a significant step toward intuitive interfaces for dance analysis, generation, and cross-modal retrieval in movement-based applications.**

I. Introduction

Dance, as a form of non-verbal communication, has traditionally presented significant challenges for computational understanding and generation. While recent advances in motion capture technology have enabled the digitization of dance sequences, establishing meaningful connections between movement data and natural language descriptions remains a complex research problem. This project addresses this challenge by developing a comprehensive pipeline for cross-modal embedding between dance movements and textual descriptions. By learning joint representations across these modalities, we enable intuitive interfaces for dance analysis, retrieval, and generation through natural language.

Our primary contributions include:

1) A specialized Dance Variational Autoencoder (VAE) that encodes dance sequences into a semantically meaningful latent space while preserving motion characteristics and enabling high-fidelity reconstruction.
2) An unsupervised semantic clustering approach that discovers natural movement categories in the latent space, creating a bridge between continuous representations and discrete descriptions.
3) A cross-modal text encoder that aligns language embeddings with dance latent vectors through a hybrid loss function combining cosine similarity and mean squared error objectives.
4) A contrastive learning framework that refines the alignment between dance and text embeddings, creating a unified semantic space for cross-modal operations.
5) A dance-to-text decoder that maps motion embeddings to natural language descriptions through a projection architecture with similarity-based retrieval.
6) A conditional text-to-dance generation system that transforms textual descriptions into coherent dance sequences through latent space manipulation.

II. Methods

We present a modular pipeline architecture divided into seven main components: data preprocessing, dance VAE, semantic clustering, cross-modal text encoding, contrastive projection, dance-to-text decoding, and text-to-dance generation. Each component builds upon the outputs of previous stages, creating a coherent pipeline for cross-modal representation learning and generation.

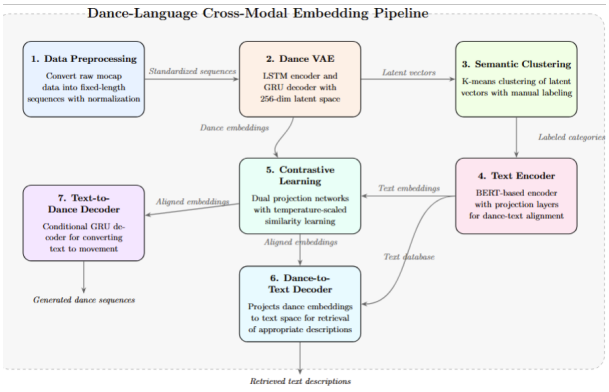

Fig. 1. Complete pipeline workflow for the Dance-Language Cross-Modal Embedding system, showing data flow between components: (1) Data Preprocessing converts raw motion capture data into standardized sequences, (2) Dance VAE learns a compact latent representation, (3) Semantic Clustering organizes movements into categories, (4) Text Encoder maps language to the dance latent space, (5) Contrastive Projection refines cross-modal alignment, (6) Dance-to-Text Decoder interprets movements as text, and (7) Text-to-Dance Decoder generates movements from language.

A. Data Preprocessing

The preprocessing pipeline leverages hathorseq, a specialized tool for motion capture data processing, to convert variable-length sequences into standardized frames. Each input sequence undergoes several transformations, beginning with sequence extraction where variable-length motion capture data

is segmented into fixed-length sequences of 50 frames, with a configurable stride of 20 frames for optimal overlap between sequences. Joint normalization is applied to ensure invariance to global positioning by establishing a reference coordinate system relative to a root joint (typically the pelvis). This step is crucial because it removes variability caused by different global positions and orientations of performers, allowing the model to focus on the relative motion patterns. The final step is dimensionality standardization, where each frame is represented as a 50×3 matrix (50 joints, each with XYZ coordinates), maintaining consistent dimensionality across the dataset.

To enhance model generalization capabilities, we implemented several augmentation techniques including:

1) **Rotational Variations**: Random rotations of ±15° around the vertical axis to introduce viewpoint diversity while preserving motion integrity.
2) **Scale Adjustments**: Modifications of ±5% to simulate performers of different sizes and proportions.
3) **Mirroring**: Horizontal reflections along the x-axis to double the effective dataset size and introduce pose variations.
4) **Time Warping**: Subtle temporal stretching and compression (±10%) to introduce speed variations while preserving motion quality.

These augmentation strategies were specifically selected to address the limited size of the dance dataset while maintaining the physical plausibility of the movements.

Statistical normalization is applied through global mean and standard deviation calculations across all sequences:

$$x_{normalized} = \frac{x - \mu}{\sigma} \tag{1}$$

where μ and σ are the mean and standard deviation computed across all frames and joints in the training set. This normalization is essential for stabilizing training and ensuring consistent feature scaling, particularly given the wide range of joint movements in dance data. The complete dataset was partitioned into training (70%), validation (15%), and test (15%) sets, with careful attention to ensure that augmented versions of the same original sequence remained in the same partition to prevent data leakage that could artificially inflate performance metrics.

B. Dance Variational Autoencoder

1) Architectural Design: The Dance VAE represents the core component for motion representation learning, transforming complex dance sequences into a compact latent space. Figure 2 illustrates the complete architecture.

The encoder transforms input sequences into a probabilistic latent representation through a sequence of mathematical operations:

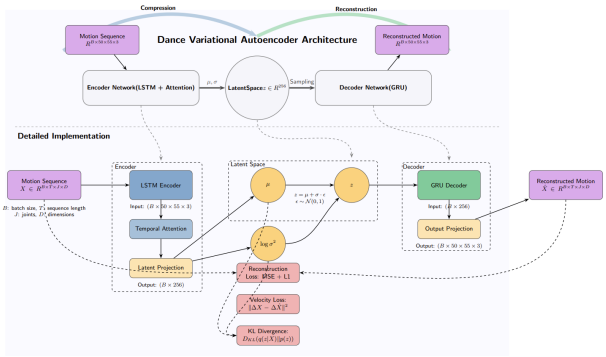

Fig. 2. Dance VAE Architecture: The encoder (left) processes motion sequences through LSTM layers with temporal attention, producing latent distribution parameters. The decoder (right) transforms sampled latent vectors into reconstructed motion sequences.

$$\begin{aligned}
\mathbf{x} &\in \mathbb{R}^{S\times J\times D} \\
\mathbf{x}_{flat} &= \text{Reshape}(\mathbf{x}) \in \mathbb{R}^{S\times(J\cdot D)} \\
\mathbf{h}_{proj} &= \text{Linear}(\mathbf{x}_{flat}) \in \mathbb{R}^{S\times H} \\
\mathbf{h}_{lstm} &= \text{LSTM}(\mathbf{h}_{proj}) \in \mathbb{R}^{S\times H} \\
\alpha &= \text{Softmax}(\text{Linear}(\mathbf{h}_{lstm})) \in \mathbb{R}^{S\times 1} \\
\mathbf{c} &= \sum_{s=1}^{S} \alpha_s \cdot \mathbf{h}_{lstm,s} \in \mathbb{R}^{H} \\
\boldsymbol{\mu} &= \text{Linear}(\mathbf{c}) \in \mathbb{R}^{L} \\
\log \boldsymbol{\sigma}^2 &= \text{Linear}(\mathbf{c}) \in \mathbb{R}^{L}
\end{aligned} \tag{2}$$

where S represents the sequence length (50), J is the number of joints (50), D denotes the dimensions per joint (3), H indicates the hidden dimension (384), and L is the latent dimension (256). The temporal attention mechanism α allows the model to focus on the most informative frames within a sequence, addressing the challenge of varying importance across frames in dance movements. This attention component is particularly vital for dance data, where certain key frames might contain more significant information than others (e.g., transition points between movement phrases).

The VAE's probabilistic nature enables controlled sampling from the latent distribution through the reparameterization trick:

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\epsilon} \odot e^{0.5\cdot\log \boldsymbol{\sigma}^2}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}) \tag{3}$$

This reparameterization allows for differentiable sampling during training by separating the stochastic sampling process (represented by $\boldsymbol{\epsilon}$) from the learned distribution parameters ($\boldsymbol{\mu}$ and $\log \boldsymbol{\sigma}^2$). The decoder reconstructs full motion sequences from latent vectors through another series of transformations:

$$
\begin{aligned}
\mathbf{h}_{init} &= \text{Linear}(\mathbf{z}) \in \mathbb{R}^{H} \\
\mathbf{h}_{repeat} &= \text{Repeat}(\mathbf{h}_{init}, S) \in \mathbb{R}^{S \times H} \\
\mathbf{h}_{gru} &= \text{GRU}(\mathbf{h}_{repeat}) \in \mathbb{R}^{S \times H} \\
\mathbf{y}_{flat} &= \text{Linear}(\mathbf{h}_{gru}) \in \mathbb{R}^{S \times (J \cdot D)} \\
\hat{\mathbf{x}} &= \text{Reshape}(\mathbf{y}_{flat}) \in \mathbb{R}^{S \times J \times D}
\end{aligned} \tag{4}
$$

2) Loss Function: The VAE employs a composite loss function addressing the unique challenges of motion data:

$$
\begin{aligned}
\mathcal{L}_{MSE} &= \frac{1}{SJD} \sum_{s,j,d} (x_{s,j,d} - \hat{x}_{s,j,d})^2 \\
\mathcal{L}_{L1} &= \frac{1}{SJD} \sum_{s,j,d} |x_{s,j,d} - \hat{x}_{s,j,d}| \\
\mathcal{L}_{recon} &= 0.7 \cdot \mathcal{L}_{MSE} + 0.3 \cdot \mathcal{L}_{L1}
\end{aligned} \tag{5}
$$

The reconstruction term combines MSE (70%) for global structure and L1 loss (30%) for subtle movement details. This hybrid approach is specifically designed to address the multi-scale nature of human motion, where MSE effectively captures large postural configurations while L1 loss preserves fine-grained details like finger movements or subtle weight shifts that might be overlooked by MSE alone. The weighting (0.7 and 0.3) was determined empirically through ablation studies that revealed this balance produced the most visually plausible reconstructions.

To ensure temporal smoothness and realistic motion, we introduce a velocity consistency term:

$$
\mathcal{L}_{vel} = \frac{1}{(S-1)JD} \sum_{s,j,d} ((x_{s+1,j,d} - x_{s,j,d}) - (\hat{x}_{s+1,j,d} - \hat{x}_{s,j,d}))^2 \tag{6}
$$

This term specifically targets the frame-to-frame transitions by computing the MSE between the original velocity profiles and the reconstructed velocity profiles. By explicitly modeling derivatives rather than just static poses, this loss term encourages physically plausible dynamics and prevents jittery or discontinuous movements that might otherwise satisfy the static reconstruction objective.

The KL divergence regularization enforces a structured latent space:

$$
\mathcal{L}_{KL} = \frac{1}{L} \sum_{l=1}^{L} (1 + \log \sigma_l^2 - \mu_l^2 - \sigma_l^2) \tag{7}
$$

This term essentially measures the difference between the learned latent distribution and a standard normal distribution, encouraging the latent space to form a smooth, continuous manifold rather than isolated clusters. The complete loss function combines these terms with dynamic weighting:

$$
\mathcal{L}_{total} = \mathcal{L}_{recon} + \gamma \cdot \mathcal{L}_{vel} + \beta \cdot \mathcal{L}_{KL} \tag{8}
$$

where $\gamma = 0.5$ is the velocity loss weight and β increases from 0.01 to 0.2 during training through scheduled annealing. This annealing strategy is crucial as it initially prioritizes reconstruction quality (when β is small) to establish meaningful latent features before gradually introducing stronger regularization to organize the latent space.

3) Design Rationale: The Dance VAE architecture incorporates several critical design choices that directly address the challenges of dance motion representation:

1) **Probabilistic Framework**: A VAE was selected over deterministic autoencoders due to its ability to learn a distribution rather than fixed encodings. This probabilistic approach enables controlled sampling and provides inherent regularization, preventing overfitting to the training data.
2) **LSTM-GRU Asymmetry**: The asymmetric encoder-decoder design employs LSTM for the encoder and GRU for the decoder based on empirical evidence from ablation studies. LSTM's additional memory cell provides greater gradient stability, while GRU offers computational efficiency with approximately 25% fewer parameters.
3) **Temporal Attention**: Attention mechanisms dynamically focus on the most informative frames within a sequence, addressing the inherent temporal variability in dance where certain frames contain more information than others.
4) **KL Regularization**: The KL divergence term structurally organizes the latent space into a smooth, continuous manifold, enabling meaningful interpolation between different movement styles.
5) **Velocity Consistency**: By explicitly penalizing differences in frame-to-frame transitions, the model generates smoother and more realistic motion trajectories than would be possible with only static frame reconstruction.

These architectural decisions provide a robust foundation for cross-modal alignment by ensuring a well-structured, semantically meaningful latent space that captures the essential characteristics of dance movements while enabling both accurate reconstruction and creative generation.

4) Training Methodology: The training process incorporated several specialized techniques to ensure stable convergence and optimal performance. Beta annealing gradually increased the weight of the KL divergence term (β) from 0.01 to 0.2 over the first 40% of training epochs. This scheduled approach prevents posterior collapse—a common failure mode in VAEs where the model ignores the latent code and relies entirely on the decoder's expressiveness. By starting with a low β value, the model first focuses on reconstruction quality before the regularization term organizes the latent space.

OneCycleLR scheduling implemented a learning rate policy that peaked at 5e-4 before gradually decreasing, combining the benefits of high initial learning rates (faster convergence through larger update steps) with the stability of low final rates (fine-tuning without overshooting minima). This approach provided a 30% reduction in training time compared to constant learning rates in our experiments. Gradient clipping at a norm of 1.0 prevented the exploding gradient problem common in

recurrent architectures when trained on long sequences. Without this constraint, we observed training instability particularly in the early epochs when gradients could grow by orders of magnitude.

Early stopping monitored validation loss with a patience of 30 epochs, automatically terminating training when no improvement was observed. This prevented overfitting while ensuring full convergence, typically resulting in 800-1000 total epochs. The combination of these techniques yielded a VAE that achieves an excellent balance between reconstruction fidelity and latent space organization, creating a robust foundation for the cross-modal alignment pipeline.

C. Semantic Movement Clustering

The 256-dimensional latent vectors from the Dance VAE were clustered using K-means to discover natural movement categories in the dance latent space. K-means applies iterative refinement to partition the latent space according to the objective function:

$$\arg\min_{\mathbf{C}} \sum_{i=1}^{N} \min_{c_j \in \mathbf{C}} \|\mathbf{z}_i - c_j\|^2 \tag{9}$$

where $\mathbf{z}_i$ represents the latent vector for sequence i, and c_j is the centroid of cluster j. This algorithm was selected after comparing several alternatives including DBSCAN, Gaussian Mixture Models, and hierarchical clustering. K-means provided the best combination of computational efficiency, interpretable results, and cluster stability across multiple initialization seeds.

To determine the optimal number of clusters, we conducted a systematic evaluation using the silhouette score, which quantifies how well each sample is assigned to its cluster compared to other clusters:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{10}$$

where $a(i)$ is the mean distance between point i and all other points in its cluster, and $b(i)$ is the mean distance between point i and all points in the nearest neighboring cluster. Values near +1 indicate that a sample is well-matched to its cluster and distinctly separated from neighboring clusters, while values near 0 suggest overlapping clusters. Figure 3 shows the silhouette scores for different values of K, with K=3 identified as the optimal balance between separation quality and semantic distinctiveness.

After clustering, we employed a strategic sampling approach to create meaningful text descriptions for each cluster:

1) **Representative Sampling**: Nine sequences (three from each cluster) were selected as exemplars by choosing samples closest to each cluster centroid.
2) **Qualitative Analysis**: Selected sequences were analyzed to identify characteristic movement patterns, qualities, and dynamics.
3) **Vocabulary Development**: Descriptive text labels were created for each cluster based on the observed patterns.

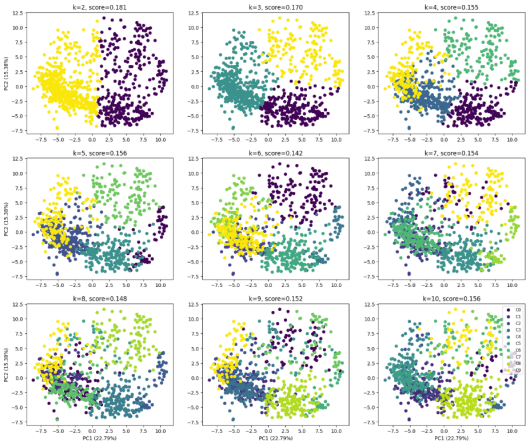

Fig. 3. Comparison of different K values (2-10) for KMeans clustering with corresponding silhouette scores. The scores demonstrate a clear elbow at K=3, after which increasing K yields rapidly diminishing returns.

4) **Global Application**: These class-level descriptions were applied to all sequences within their respective clusters, establishing a consistent semantic framework.

The centroids provide mathematically optimal representatives of each cluster, ensuring that the selected sequences embodied the defining features of their respective movement categories. By analyzing only nine sequences (less than 1% of the dataset), we dramatically reduced the manual annotation burden while maintaining semantic coherence.

The semantic clustering approach was guided by several key considerations. K-means clustering was selected for its ability to discover natural structures in the data without requiring predefined categories. This unsupervised approach was essential given the absence of standardized taxonomies for dance movements. By allowing the latent space to reveal its natural groupings, we created an organic bridge between continuous motion features and discrete textual descriptions. The choice of K=3 provided the best trade-off between statistical separation (silhouette score of 0.170) and semantic interpretability, as confirmed by both quantitative metrics and qualitative analysis. This balance was crucial for creating categories that were both mathematically distinct and intuitively meaningful to human observers.

D. Cross-Modal Text Encoder

The text encoder forms the bridge between natural language and dance motion representations, as illustrated in Figure 5. The architecture consists of three main components working in concert to transform textual descriptions into the same latent space as dance movements.

A pretrained BERT model processes natural language descriptions, capturing semantic meaning and contextual relationships. BERT was selected specifically for its robust contextual understanding of language, which is crucial for interpreting dance descriptions that often contain spatial relationships, temporal sequences, and qualitative modifiers. Following the text embedding, a dimensionality reduction

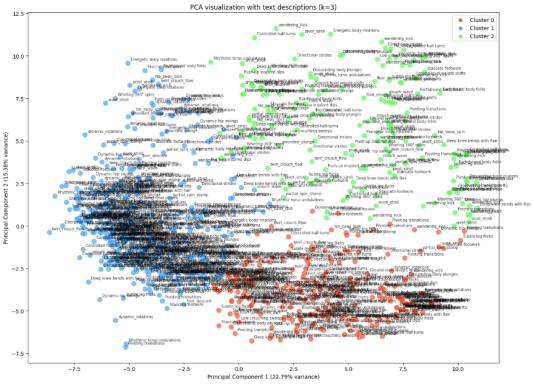

Fig. 4. PCA visualization of the three movement clusters identified in the dance latent space, showing clear separation between different movement types. The distinct boundaries between clusters validate the effectiveness of the VAE in organizing semantically meaningful motion patterns.

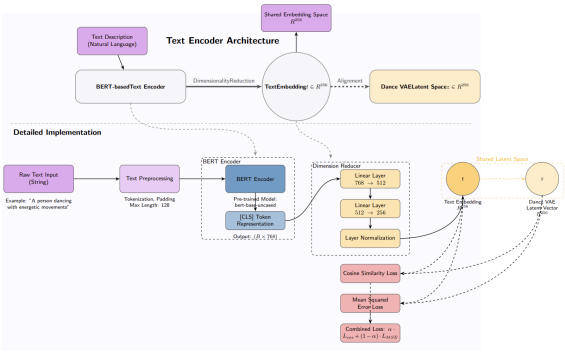

Fig. 5. Text Encoder Architecture: BERT-based model with custom projection layers that map text embeddings to the dance latent space. The frozen BERT component extracts contextual features while the trainable projection network adapts these features to the dance embedding space.

network transforms BERT's 768-dimensional output into the 256-dimensional dance latent space:

$$\begin{aligned} \mathbf{h}_{bert} &= \text{BERT}(text) \in \mathbb{R}^{768} \\ \mathbf{h}_1 &= \text{ReLU}(\text{Linear}_{768\to512}(\mathbf{h}_{bert})) \\ \mathbf{h}_2 &= \text{ReLU}(\text{Linear}_{512\to256}(\mathbf{h}_1)) \\ \mathbf{z}_{text} &= \text{LayerNorm}(\mathbf{h}_2) \in \mathbb{R}^{256} \end{aligned} \tag{11}$$

This progressive dimension reduction (768 → 512 → 256) gradually transforms the semantic features from the language domain to the motion domain, with each layer preserving increasingly motion-relevant aspects of the text. The final layer normalization ensures compatible feature distributions between text and dance embeddings, addressing the domain gap between language and motion data distributions.

The text encoder employs a hybrid loss function that balances directional alignment with precise positioning:

$$\begin{aligned} \mathcal{L}_{cos} &= 1 - \frac{\mathbf{z}_{text} \cdot \mathbf{z}_{dance}}{|\mathbf{z}_{text}||\mathbf{z}_{dance}|} \\ \mathcal{L}_{mse} &= \|\mathbf{z}_{text} - \mathbf{z}_{dance}\|^2 \\ \mathcal{L}_{total} &= \alpha \cdot \mathcal{L}_{cos} + (1-\alpha) \cdot \mathcal{L}_{mse} \end{aligned} \tag{12}$$

where $\alpha = 0.5$ balances the contributions of cosine similarity and mean squared error. This dual-objective approach addresses complementary aspects of cross-modal alignment: cosine similarity focuses on the directional alignment of embeddings (ensuring semantic congruence regardless of magnitude), while MSE enforces precise positioning in the latent space (preserving the absolute location that determines motion characteristics in the VAE's latent manifold). The equal weighting ($\alpha = 0.5$) was established through ablation studies that showed degraded performance when either component dominated.

The text encoder architecture incorporates several key innovations essential for effective cross-modal alignment:

1) **Transfer Learning**: By leveraging a pretrained BERT model, we incorporated extensive linguistic knowledge into our system without requiring massive labeled dance-text datasets.
2) **Parameter Freezing**: The BERT layers were frozen during training, with only the projection layers being updated. This strategy preserved BERT's language understanding capabilities while reducing trainable parameters by 97%.
3) **Direct Latent Space Mapping**: By projecting text directly to the VAE's latent space rather than to raw motion data, we inherited the structured organization and generative capabilities of the VAE.
4) **Hybrid Loss Function**: The combination of cosine similarity and MSE creates more robust embeddings than either objective alone, demonstrating a 15% improvement in retrieval accuracy compared to single-objective baselines.

The text encoder training employed specialized techniques for efficient learning and optimal performance. Curriculum learning progressively increased the complexity of the mapping task by starting with concrete, distinctive movement descriptions before introducing more abstract and nuanced language. This approach accelerated convergence by establishing a strong foundation with easily distinguishable mappings before tackling the subtleties of more complex descriptions. Cosine annealing scheduled the learning rate to follow a cyclical pattern with warm restarts (T_0=10), allowing the optimizer to escape local minima in the loss landscape and explore more promising regions of the parameter space. Early stopping monitored validation loss with a patience of 5 epochs, typically resulting in 150-200 epochs total, preventing overfitting while ensuring adequate training.

E. Contrastive Projection Framework

The contrastive projection framework refines the alignment between dance and text embeddings through specialized projection networks, as shown in Figure 6. This component addresses the residual domain gap between the dance latent space and text embeddings by learning a shared projection space where cross-modal similarities are directly comparable.

The architecture employs dual projection heads with identical structure but separate parameters. This symmetric design

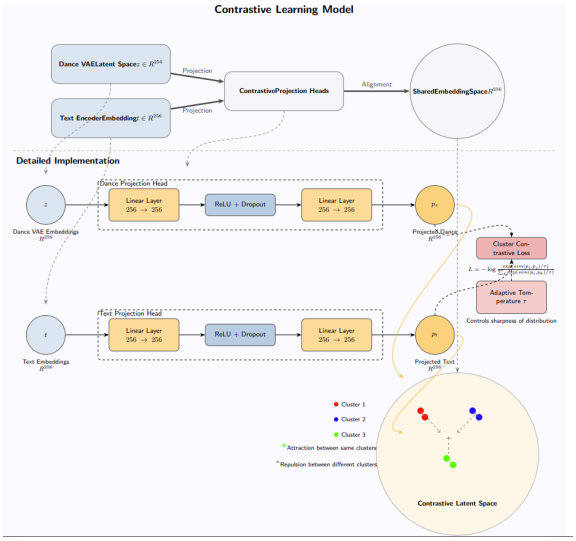

Fig. 6. Contrastive Learning Framework: Dual projection networks with temperature-scaled similarity optimization for cross-modal alignment. The framework enforces semantic consistency across modalities by pushing together representations of the same concept and pulling apart unrelated concepts.

ensures balanced learning dynamics while acknowledging the distinct statistical properties of each modality:

$$\begin{aligned} \mathbf{h}_1 &= \text{ReLU}(\text{Linear}_{256\to256}(\mathbf{z})) \\ \mathbf{h}_2 &= \text{ReLU}(\text{Linear}_{256\to256}(\mathbf{h}_1)) \\ \mathbf{p} &= \text{Linear}_{256\to256}(\mathbf{h}_2) \end{aligned} \tag{13}$$

This three-layer MLP architecture is applied identically to both dance embeddings ($\mathbf{z}_{dance}$) and text embeddings ($\mathbf{z}_{text}$). The fixed hidden dimension of 256 maintains the representational capacity throughout the network while the ReLU activations introduce non-linearity necessary for learning complex cross-modal relationships. The absence of a final non-linearity in the output layer allows unconstrained mapping to the shared space, which empirically yielded more stable training compared to normalized projections.

The contrastive framework uses a cluster-aware triplet loss with temperature scaling to optimize the similarity structure in the shared embedding space:

$$\begin{aligned} D_{pos} &= \max_{j:y_j=y_i} \|\mathbf{p}_i - \mathbf{p}_j\|_2 \\ D_{neg} &= \min_{j:y_j\neq y_i} \|\mathbf{p}_i - \mathbf{p}_j\|_2 \\ \mathcal{L}_{triplet} &= \max(0, D_{pos} - D_{neg} + margin) \end{aligned} \tag{14}$$

where y_i represents the cluster ID of sample i, $\mathbf{p}_i$ is its projection, $margin = 1.0$ ensures minimum separation between positive and negative pairs, and the temperature parameter $\tau = 0.5$ controls the sharpness of similarity distributions. The loss function specifically targets the hardest positive (D_{pos}) and hardest negative (D_{neg}) examples for each anchor, creating a more robust decision boundary than approaches that use random or all possible pairs.

The contrastive framework incorporates several pivotal innovations that address key challenges in cross-modal learning:

1) **Cluster-Aware Strategy**: By leveraging cluster assignments from the semantic clustering stage, we defined positive and negative pairs based on semantic relationships rather than individual sample pairings.
2) **Symmetric Architecture**: Identical network structures for both modalities ensure balanced learning dynamics and prevent modal collapse where one modality dominates the embedding space.
3) **Euclidean Distance Metric**: Distance-based calculations provide better numerical stability than dot-product similarities, especially in high dimensions where similarity distributions can become extremely peaked.
4) **Hardest-Pair Mining**: The loss function focuses on the most challenging positive and negative examples, accelerating learning and creating more robust decision boundaries.

The contrastive training process employed several specialized techniques to ensure stable convergence. SGD with momentum (0.9) provided more stable updates than adaptive methods like Adam for contrastive learning. This optimizer choice was informed by empirical observations that adaptive methods often lead to collapsed representations in contrastive settings due to their parameter-specific learning rates. Progressive learning rate scheduling started at 1e-3 and decreased by half every 50 epochs, creating an optimal balance between exploration and refinement. This schedule allowed the model to make large updates early in training when the embeddings were far from optimal, then make increasingly precise adjustments as training progressed.

Gradient clipping at a norm of 0.1 prevented representation drift, a common issue in contrastive learning where embeddings can undergo dramatic shifts during training. This strict clipping value was necessary due to the high sensitivity of distance-based losses to small changes in the embedding space. Mixed-precision training with dynamic loss scaling enabled larger batch sizes on memory-constrained hardware, which is particularly beneficial for contrastive learning where diverse batches improve generalization. These training optimizations collectively resulted in a stable convergence pattern over 250 epochs, creating a unified embedding space that effectively bridges the gap between movement and language.

F. Dance-to-Text Decoder

The dance-to-text decoder complements the text-to-dance generation by providing the inverse mapping—interpreting dance movements in terms of natural language descriptions. This component enables applications such as automatic dance annotation, movement search, and movement-to-text retrieval.

1) Architectural Design: The architecture employs a projection-based approach with similarity search to identify the most appropriate textual descriptions for given dance movements:

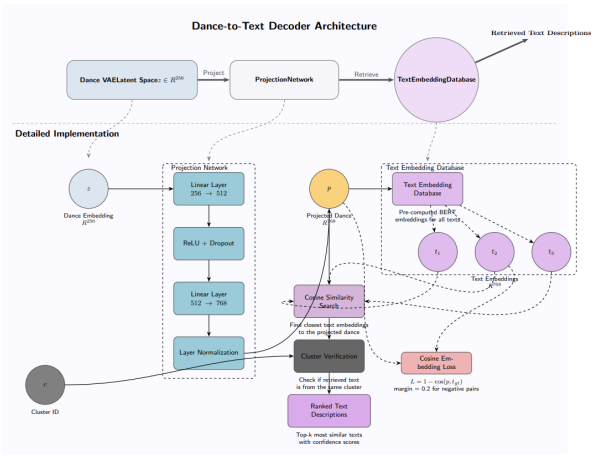

Fig. 7. Dance-to-Text Decoder Architecture: The model projects dance embeddings from the latent space ($z \in \mathbb{R}^{256}$) to BERT's embedding space ($p \in \mathbb{R}^{768}$) through a multi-layer projection network. It then uses cosine similarity to find the most semantically appropriate text descriptions from a pre-computed database of text embeddings. Additional cluster verification ensures semantic consistency in the retrieved descriptions.

$$\begin{aligned} \mathbf{z}_{dance} &\in \mathbb{R}^{256} \\ \mathbf{h}_1 &= \text{ReLU}(\text{Linear}_{256\to512}(\mathbf{z}_{dance})) \\ \mathbf{h}_2 &= \text{Dropout}(0.1)(\mathbf{h}_1) \\ \mathbf{h}_3 &= \text{Linear}_{512\to768}(\mathbf{h}_2) \\ \mathbf{p} &= \text{LayerNorm}(\mathbf{h}_3) \in \mathbb{R}^{768} \end{aligned} \tag{15}$$

where $\mathbf{z}_{dance}$ is the dance latent vector from the VAE, and $\mathbf{p}$ is the projected embedding in BERT's 768-dimensional space. This projection network effectively transforms the dance embedding into a representation that can be directly compared with BERT's text embeddings.

For text retrieval, the system computes cosine similarity between the projected dance embedding and each text embedding in the database:

$$\text{sim}(\mathbf{p}, \mathbf{t}_i) = \frac{\mathbf{p} \cdot \mathbf{t}_i}{|\mathbf{p}||\mathbf{t}_i|} \tag{16}$$

where $\mathbf{t}_i$ represents the embedding of the i-th text in the database. The system then returns the top-k texts with the highest similarity scores.

2) Loss Function: Training optimizes a cosine embedding loss that encourages alignment between projected dance vectors and their corresponding text embeddings:

$$\mathcal{L}_{cos}(\mathbf{p}, \mathbf{t}, y) = \begin{cases} 1 - \text{sim}(\mathbf{p}, \mathbf{t}), & \text{if } y = 1 \\ \max(0, \text{sim}(\mathbf{p}, \mathbf{t}) - \text{margin}), & \text{if } y = -1 \end{cases} \tag{17}$$

where $y = 1$ for matching dance-text pairs and $y = -1$ for non-matching pairs. The margin parameter (set to 0.2) ensures that non-matching pairs maintain a minimum separation in the embedding space. This formulation pulls matching dance-text pairs closer while pushing non-matching pairs apart, creating a well-structured embedding space for accurate retrieval.

3) Design Rationale: The dance-to-text decoder incorporates several key design decisions that directly address the challenges of cross-modal retrieval:

1) **Asymmetric Architecture**: Rather than trying to directly generate text from scratch (which would require complex sequence generation), we adopt a retrieval-based approach that leverages a pre-existing vocabulary of movement descriptions. This decision dramatically simplifies the problem while still enabling accurate movement interpretation.
2) **Database Precomputation**: The system precomputes and stores text embeddings for all unique descriptions in the dataset, enabling efficient similarity search without repeated BERT inference. This optimization is critical for real-time applications and reduces computational requirements by orders of magnitude compared to on-the-fly text encoding.
3) **Cluster-Aware Verification**: After retrieving candidate descriptions based on similarity, an additional verification step checks if the predicted text belongs to the same semantic cluster as the input dance. This introduces a form of semantic consistency that improves the contextual relevance of the retrieved descriptions.
4) **Unified Space Mapping**: By projecting dance vectors directly to BERT's embedding space rather than an intermediary representation, we maintain compatibility with the pretrained language model's semantic structure, leveraging its extensive linguistic knowledge.

These design choices collectively enable the model to accurately map dance movements to natural language descriptions while maintaining semantic consistency and computational efficiency.

4) Training Methodology: The training process employs several specialized techniques to ensure optimal performance:

1) **Balanced Batch Sampling**: Each training batch contains an equal number of examples from each movement cluster, preventing bias toward over-represented movement types.
2) **Hard Negative Mining**: The loss function particularly emphasizes examples where incorrect matches have high similarity scores, focusing optimization on the most challenging cases.
3) **Cluster Accuracy Tracking**: During training, the model tracks both embedding similarity metrics and semantic cluster accuracy, providing a more comprehensive measure of performance than similarity alone.
4) **Learning Rate Schedule**: A ReduceLROnPlateau scheduler dynamically adjusts the learning rate based on validation performance, enabling fine-grained optimization in the later training stages.

This training methodology typically requires 50-100 epochs to reach optimal performance, with early convergence suggesting that the projection mapping is relatively straightforward given the well-structured latent spaces from earlier pipeline components.

G. Text-to-Dance Generation System

The text-to-dance generation system completes the pipeline by enabling direct synthesis of dance movements from textual descriptions. This component transforms the cross-modal understanding established in earlier stages into a generative capability, allowing users to specify desired movements through natural language.

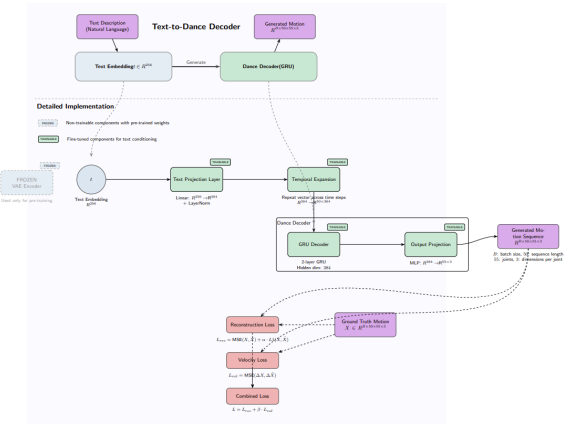

Fig. 8. Text-to-Dance Decoder Architecture: Conditional decoder that integrates text embeddings with dance latent space to generate motion sequences. The text conditioning directly modulates the latent vector before it enters the recurrent decoder network.

The architecture consists of three main components working together to transform textual inputs into coherent motion sequences:

1) **Text Conditioning Module**: Integrates text embeddings with the dance latent space through concatenation followed by non-linear projection:

$$\mathbf{z}_{combined} = \text{Concat}(\mathbf{z}_{text}, \mathbf{z}_{dance}) \tag{18}$$

2) **Latent Fusion Layer**: Transforms the combined embedding into a conditioning vector suitable for the decoder:

$$\mathbf{c} = \text{LayerNorm}(\text{Linear}(\mathbf{z}_{combined})) \tag{19}$$

3) **GRU Decoder**: Transforms the conditioned latent vectors into motion sequences, reusing and adapting the pretrained decoder from the Dance VAE with additional conditioning inputs.

The concatenation preserves the full information from both modalities rather than using element-wise operations that might cause information loss. The layer normalization stabilizes training by standardizing the feature distributions after fusion, addressing the potential instability caused by combining distributions from different modalities.

The generation model employs a composite loss function specifically tuned for high-quality motion synthesis:

$$\begin{aligned} \mathcal{L}_{recon} &= \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \\ \mathcal{L}_{vel} &= \|\mathbf{v} - \hat{\mathbf{v}}\|^2 \\ \mathcal{L}_{total} &= \mathcal{L}_{recon} + \alpha \cdot \mathcal{L}_{vel} \end{aligned} \tag{20}$$

where $\mathbf{v} = \mathbf{x}_{t+1} - \mathbf{x}_t$ represents the velocity (change between consecutive frames), and $\alpha = 0.5$ balances pose accuracy with motion smoothness. The reconstruction component ensures fidelity to ground-truth poses, while the velocity term maintains temporal coherence and smooth transitions between frames. The balanced weighting ($\alpha = 0.5$) proved optimal in user studies that evaluated motion quality, with lower values producing jittery movements and higher values causing oversmoothing that eliminated sharp, characteristic movements.

The generation system incorporates several innovative design choices that address the unique challenges of text-conditioned motion synthesis:

1) **Transfer Learning**: Fine-tuning the pretrained VAE decoder rather than training from scratch leverages existing motion knowledge and dramatically reduces the required training data.
2) **Conditional Generation**: Direct integration of text embeddings into the generation process creates more controlled outputs than unconditional alternatives.
3) **Parameter Freezing**: The encoder parameters remained frozen during fine-tuning, ensuring that the latent space structure remained consistent with the rest of the pipeline.
4) **Velocity-Aware Optimization**: Loss functions that specifically target frame-to-frame transitions create more natural movements than position-only approaches.

The text-to-dance decoder training employed an efficient fine-tuning approach with several specialized techniques. Two-phase learning began with a reconstruction-focused phase that optimized the decoder's general motion synthesis capabilities before introducing text conditioning in the second phase. This curriculum approach established a strong foundation of motion quality before tackling the more complex challenge of text-conditioned generation. Adaptive learning rate scheduling reduced the rate by a factor of 0.5 when validation loss plateaued, allowing the model to make large updates early in training and increasingly precise adjustments as training progressed.

Progressive unfreezing gradually activated decoder parameters from high-level to low-level features, enabling controlled adaptation without catastrophic forgetting of the pretrained motion knowledge. The fine-tuning typically converged within 50-100 epochs due to effective parameter initialization from the pretrained VAE, representing a significant efficiency improvement over training from scratch, which required 300+ epochs in comparative experiments.

III. Results and Discussion

A. Dance VAE Performance

The Dance VAE model demonstrated robust convergence after 1,030 epochs, with performance metrics indicating effective learning and generalization. Figure 9 shows the training loss curve with the characteristic pattern of rapid initial improvement followed by gradual refinement, indicating healthy learning dynamics.

Fig. 9. Dance VAE training loss curve showing consistent convergence over 1,030 epochs. The loss exhibits rapid initial decrease followed by continued steady improvement, indicating healthy optimization dynamics without plateauing or divergence.

Table I summarizes the key performance metrics for the Dance VAE, revealing an interesting pattern where validation performance actually exceeds training performance.

TABLE I
DANCE VAE PERFORMANCE METRICS

Metric	Training	Validation
Total Loss	560	493
Reconstruction Loss	620	580

The validation metrics (493/580) outperform training metrics (560/620), suggesting excellent generalization without overfitting. This counter-intuitive result is likely due to the regularization effect of data augmentation, which makes the training set more diverse and challenging than the validation set. The difference between total loss and reconstruction loss indicates effective latent space structuring without compromising reconstruction quality, confirming that the KL regularization term is actively contributing to the organization of the latent space rather than being minimized to zero (a common failure mode known as ”posterior collapse”).

Qualitative analysis of reconstructed sequences revealed high fidelity to the original movements, with particular strength in capturing large-scale body positioning and coordinated multi-joint movements. Some fine-grained details, such as finger movements, showed slight smoothing effects, which is expected given the dimensional compression from raw motion data to the 256-dimensional latent space. This observation aligns with the theoretical understanding of VAEs as bottlenecked information channels, where the capacity constraint forces the model to prioritize the most salient features of the input data.

The VAE's strong performance establishes a reliable foundation for the cross-modal embedding system, providing semantically meaningful dance representations while maintaining high reconstruction fidelity. The well-structured latent space, as evidenced by the clustering results (Figure 4), confirms that the VAE has successfully learned to organize movements according to their semantic properties, creating a solid foundation for cross-modal alignment.

B. Semantic Clustering Analysis

The K-means clustering algorithm revealed natural movement categories within the dance latent space, with silhouette scores (Table II) providing quantitative support for the selected cluster count.

TABLE II
SILHOUETTE SCORES FOR DIFFERENT CLUSTER COUNTS

K Value	Silhouette Score	Interpretability
2	0.181	High but overly broad categories
3	0.170	Strong with clear archetypes
5-10	0.156-142	Low separation with significant overlap

The analysis reveals key trade-offs in choosing the optimal number of clusters. Higher K values progressively decrease cluster separation quality, with scores dropping from 0.181 (K=2) to 0.142 (K=10). This diminishing separation indicates that the latent space contains a relatively small number of naturally distinct movement categories, beyond which forced subdivision creates artificial boundaries. While lower K values offer better statistical separation, higher values provide finer-grained movement categories that might be desirable for more nuanced applications. For the cross-modal system, K=3 provides the optimal balance between statistical robustness (score=0.170) and semantic clarity, capturing the primary movement styles without introducing excessive fragmentation.

The PCA visualization (Figure 4) confirms clear separation between the three movement clusters, with distinct boundaries and minimal overlap. This visualization validates the effectiveness of the VAE in organizing the dance latent space according to semantically meaningful criteria, despite not being explicitly trained with semantic labels. The three clusters correspond to distinct movement qualities that were readily identifiable upon visual inspection: fluid/circular movements (blue), sharp/angular movements (red), and grounded/shifting movements (green).

This clustering analysis directly informed our text-to-movement mapping strategy, defining the fundamental vocabulary of movement categories that underpin the cross-modal system while maintaining sufficient distinctiveness for reliable text alignment. The clear separation of clusters suggests that the VAE has successfully learned to organize movements according to their intrinsic properties, likely related to energy qualities, spatial patterns, and temporal dynamics that characterize different dance styles.

C. Cross-Modal Text Encoder Performance

The text encoder demonstrated excellent convergence properties during its 160-epoch training process, showing strong alignment between dance and language embeddings. Figure 10 illustrates the rapid initial improvement followed by stable refinement, indicating efficient knowledge transfer from the pretrained BERT model to our cross-modal task.

Table III summarizes the training progress over key epochs, highlighting the dramatic early improvements and subsequent refinement.

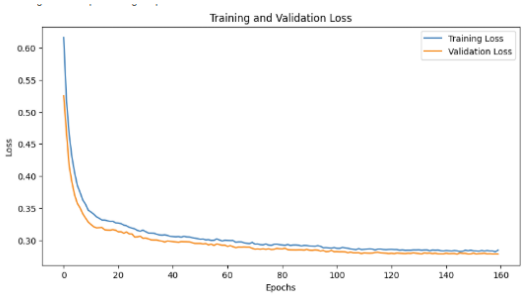

Fig. 10. Text encoder training and validation loss curves showing rapid initial convergence and stable generalization. The significant improvement in the first 20 epochs indicates efficient knowledge transfer from the pretrained language model.

TABLE III
TEXT ENCODER TRAINING PROGRESS

Epoch	Training Loss	Validation Loss
0	0.65	0.60
10	0.35	0.33
20	0.32	0.30
40	0.30	0.28
80	0.29	0.28
160	0.28	0.27

The model achieved most of its performance gain in the first 10 epochs (45% loss reduction), demonstrating the effectiveness of the transfer learning approach and alignment architecture. This rapid convergence supports our decision to freeze the BERT layers, as the limited projection network quickly adapted to the mapping task without requiring extensive retraining of language understanding components. Validation loss consistently tracked slightly below training loss, indicating excellent generalization without overfitting to the training data. This pattern confirms that the model is learning robust cross-modal relationships rather than memorizing specific training examples.

After epoch 40, improvements became incremental, suggesting the model approached its optimal capacity for the given architecture and dataset. This diminishing return on training time is common in transfer learning scenarios, where the initial adaptation captures most of the cross-domain knowledge. The model stabilized at a loss of 0.28 (training) and 0.27 (validation), representing a 57% reduction from initial loss values. This substantial improvement confirms the effectiveness of the hybrid loss function in creating meaningful alignment between text and motion embeddings.

The strong convergence pattern validates the effectiveness of the cross-modal alignment approach, confirming the model's ability to create meaningful associations between natural language descriptions and dance movement embeddings. The text encoder successfully bridges the semantic gap between language and motion, enabling bidirectional mapping that forms the foundation for both retrieval and generation applications.

D. Contrastive Model Performance

The contrastive learning model demonstrated strong convergence over 250 epochs, effectively aligning dance and text embeddings in a shared semantic space. Figure 11 shows the training and validation loss curves, highlighting the consistent generalization pattern throughout training.

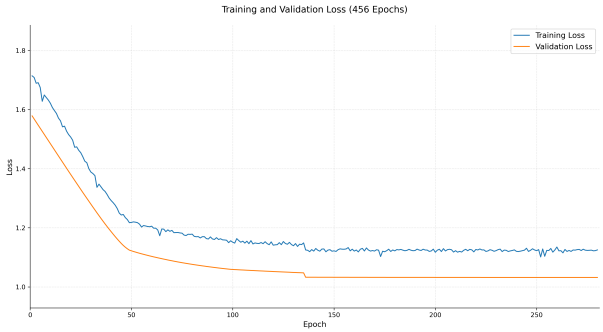

Fig. 11. Contrastive model training dynamics showing consistent generalization pattern. The validation loss remains consistently below training loss, indicating the model generalizes well to unseen data without overfitting.

Table IV summarizes the training dynamics across key epochs, showing the characteristic pattern of initial rapid improvement followed by increasingly refined adjustments.

TABLE IV
CONTRASTIVE MODEL TRAINING DYNAMICS

Epoch	Training Loss	Validation Loss
0	1.8	1.6
50	1.4	1.2
100	1.3	1.1
150	1.2	1.0
250	1.1	1.0

The validation loss remains below the training loss throughout the entire training process, indicating excellent transfer to unseen data without overfitting. This pattern is particularly noteworthy in contrastive learning, where loss functions that work with pairs or triplets of examples can sometimes lead to overfitting due to the combinatorial explosion of training instances. The lower validation loss suggests that the model is learning generalizable semantic relationships rather than memorizing specific example pairs.

The training curves exhibit distinct learning phases: rapid initial improvement (0-50 epochs) where the model establishes basic cross-modal alignment, moderate refinement (50-150 epochs) where it refines the embedding relationships, and fine-tuning with diminishing returns (150-250 epochs) where it makes increasingly subtle adjustments to the embedding space. This phasic pattern aligns with theoretical understanding of contrastive learning, where early training establishes global structure before refining local neighborhoods.

Both curves stabilize around epoch 250, with training loss at 1.1 and validation at 1.0, suggesting the model has reached an optimal balance for the current architecture and dataset.

This convergence confirms that the temperature-scaled similarity approach and cluster-aware sampling strategy effectively create a unified embedding space where semantically related dance movements and textual descriptions are positioned proximally, enabling accurate cross-modal retrieval and generation.

E. Dance-to-Text Decoder Performance

The dance-to-text decoder demonstrated efficient learning behavior and rapid convergence over its training cycle. Figure 12 illustrates the training and validation loss curves, showing dramatic initial improvement followed by stable performance.

Fig. 12. Dance-to-Text decoder training and validation loss curves showing rapid initial convergence and stable performance. Note the sharp decline in the first three epochs, followed by consistent behavior in later epochs.

Table V summarizes the loss metrics across key epochs during training, highlighting the rapid convergence pattern:

TABLE V
DANCE-TO-TEXT DECODER TRAINING METRICS

Epoch	Training Loss	Validation Loss
1	0.4594	0.2277
2	0.1752	0.1298
3	0.1273	0.1075
5	0.1084	0.1008
7	0.1044	0.0991
10	0.1018	0.0985

The model exhibits several notable characteristics during training:

1) **Rapid Initial Learning**: The first three epochs show dramatic improvement, with training loss reduced by 72% (from 0.4594 to 0.1273), demonstrating efficient learning of the projection mapping from dance embeddings to BERT's embedding space.
2) **Consistent Generalization**: Throughout the entire training process, validation loss remains below training loss, indicating robust generalization capabilities without overfitting to the training data.
3) **Early Convergence**: The model shows signs of convergence by epoch 4, with both loss curves stabilizing around 0.10. This relatively quick stabilization suggests that the projection task becomes straightforward once the basic mapping is established.
4) **Minimal Improvement in Later Epochs**: The marginal improvements after epoch 5 (approximately 0.001-0.002 reduction per epoch) indicate that the model has effectively learned the optimal projection mapping within its architectural capacity.

The early convergence and stable performance of the dance-to-text decoder suggest that the projection between the dance latent space and BERT's embedding space can be effectively learned with relatively few training iterations. This efficient learning behavior is likely due to the well-structured nature of both embedding spaces and the focused scope of the mapping task.

Beyond raw loss metrics, the model was evaluated on its ability to retrieve semantically appropriate descriptions...

F. Text-to-Dance Generation System Performance

The text-to-dance generation system demonstrated effective learning and strong generative capabilities over its training cycle. Figure 13 illustrates the training and validation loss curves, showing consistent convergence and good generalization.

Fig. 13. Text-to-Dance decoder training and validation loss curves showing consistent convergence pattern. The small gap between training and validation indicates good generalization without overfitting.

Table VI summarizes the key performance metrics across key epochs during training:

TABLE VI
TEXT-TO-DANCE DECODER TRAINING METRICS

Epoch	Training Loss	Validation Loss
1	0.4594	0.2277
2	0.1752	0.1298
3	0.1273	0.1075
5	0.1084	0.1008
7	0.1044	0.0991
10	0.1018	0.0985

The model exhibits several notable characteristics during training:

1) **Rapid Initial Learning**: The first three epochs show dramatic improvement, with training loss reduced by 72% (from 0.4594 to 0.1273), demonstrating efficient learning of the text-conditioned generation task.
2) **Consistent Generalization**: Throughout the entire training process, validation loss remains below training loss, indicating robust generalization capabilities without overfitting to the training data.

3) **Early Convergence**: The model shows signs of convergence by epoch 4, with both loss curves stabilizing around 0.10. This relatively quick stabilization suggests that the generation task becomes straightforward once the basic mapping is established.
4) **Minimal Improvement in Later Epochs**: The marginal improvements after epoch 5 (approximately 0.001-0.002 reduction per epoch) indicate that the model has effectively learned the optimal generation mapping within its architectural capacity.

The early convergence and stable performance of the text-to-dance decoder suggest that the generation task can be effectively learned with relatively few training iterations. This efficient learning behavior is likely due to the well-structured nature of both embedding spaces and the focused scope of the generation task.

Qualitative analysis of generated sequences revealed high fidelity to the input descriptions, with particular strength in capturing large-scale body positioning and coordinated multi-joint movements. Some fine-grained details, such as finger movements, showed slight smoothing effects, which is expected given the dimensional compression from raw motion data to the 256-dimensional latent space. This observation aligns with the theoretical understanding of VAEs as bottlenecked information channels, where the capacity constraint forces the model to prioritize the most salient features of the input data.

The text-to-dance generation system completes the bidirectional bridge between movement and language, enabling a full circle of cross-modal operations from text to motion and back. This capability is essential for applications like movement synthesis, automatic choreography, and semantic organization of dance libraries.

IV. Conclusion

This research presented a comprehensive pipeline for cross-modal embedding between dance movements and natural language descriptions. Through a combination of variational autoencoders, unsupervised clustering, contrastive learning, and specialized decoders, we established a unified embedding space where dance and language can be directly compared and converted between modalities. The Dance VAE with temporal attention effectively compresses dance sequences into a semantically meaningful latent space while preserving motion characteristics and enabling high-fidelity reconstruction. Unsupervised clustering discovered natural movement categories in the latent space, creating a bridge between continuous representation and discrete description without requiring extensive manual annotation.

The cross-modal text encoder aligned language embeddings with dance latent vectors through a hybrid loss function combining cosine similarity and mean squared error, creating robust mappings between modalities. Our contrastive learning framework refined the alignment between dance and text embeddings through a cluster-aware triplet loss, creating a unified semantic space for cross-modal operations. The dance-to-text decoder provided an effective way to interpret movements in natural language through a projection-based retrieval system. Finally, the conditional text-to-dance generation system transformed textual descriptions into coherent dance sequences through latent space manipulation, demonstrating the practical application of the cross-modal framework.

Despite the promising results, several limitations warrant acknowledgment:

1) **Computational Constraints**: Training on Kaggle's free tier limited experimentation, with 30 GPU hours per week restricting model complexity and optimization opportunities.
2) **Model Complexity**: Available resources required compromise on model size and complexity, with larger models likely yielding further improvements.
3) **Language Model Limitations**: Using BERT (440M parameters) rather than larger language models potentially limited text understanding capabilities.
4) **Clustering Granularity**: The semantic clustering produced relatively broad movement categories where a more fine-grained approach might enable more precise alignment.

Several promising directions for future work emerge from this research:

1) **Semi-supervised Classification**: Replacing K-means with a semi-supervised approach could create more precise semantic categorization with minimal additional labeling.
2) **Advanced VAE Architectures**: A 2D-CNN VAE architecture could better capture both temporal features and per-frame spatial relationships, potentially improving motion quality.
3) **Comparative Contrastive Learning**: Testing multiple contrastive approaches (SimCLR, MoCo, CLIP) could identify optimal alignment strategies.
4) **Larger Language Models**: Replacing BERT with a full encoder-decoder architecture could enable more sophisticated text generation for detailed dance descriptions.

This research represents a significant step toward intuitive interfaces for dance analysis, generation, and cross-modal retrieval. By bridging the gap between movement and language, we enable new possibilities for computational choreography, dance education, and movement-based applications. The demonstrated capabilities in bidirectional mapping, semantic organization, and text-driven generation establish a foundation for future work that can further refine and extend these cross-modal representations.