RIDGE: Rule-Infused Deep Learning for Realistic Co-Speech Gesture Generation

Ghazanfar Ali
Intelligence and Interaction Research Center,
Korea Institute of science and technology(KIST),
Seoul, Republic of Korea
ali@kist.re.kr

HwangYoun Kim
AI-Robotics, KIST School,
University of Science and Technology(UST),
Seoul, Republic of Korea
daqjjang@ust.ac.kr

Jae-In Hwang
Intelligence and Interaction Research Center,
Korea Institute of science and technology(KIST),
Seoul, Republic of Korea
hji@kist.re.kr

Abstract

Co-speech gestures are essential for natural human communication, yet existing synthesis methods fall short in delivering semantically aligned and contextually appropriate motions. In this paper, we present **RIDGE**, a hybrid system that combines rule-based and deep learning approaches to generate realistic gestures for virtual avatars and human-computer interaction. RIDGE employs a high-fidelity rule base generated from motion capture data with the assistance of large language models, to select reliable gesture mappings. When a high-confidence match is not available, a contrastively trained deep learning model steps in to produce semantically appropriate gestures. Evaluated using a novel Gesture Cluster Affinity (GCA) metric, our system outperforms existing baselines, achieving a GCA score of 0.73 compared to rule-based baseline 0.6 and end-to-end: 0.52, while ground truth score was 0.90. Detailed analyses of system architecture, data preprocessing, and evaluation methodologies demonstrate RIDGE's potential to enhance gesture synthesis. Project Url: `https://www.mrlab.co.kr/research/ridge`

Keywords: Co-speech Gestures, HCI, virtual worlds, computer animation

1 Introduction

Co-speech gestures are integral to natural human communication, enriching spoken language with non-verbal cues that enhance perceived realism and conveyed meaning [1]. Consequently, generating realistic and contextually appropriate gestures for virtual avatars and human-computer interaction systems is cru-

cial for effective and engaging communication. However, achieving this remains a significant challenge, with traditional gesture synthesis methods often struggling to capture the required nuances [2].

Historically, research has largely followed two paths: rule-based systems and end-to-end deep learning models. Early rule-based systems often relied on small, manually curated gesture libraries (sometimes only 50-60 units), leading to repetitive outputs [3]. While later work expanded these libraries significantly (e.g., up to 1,000 gestures with rule bases exceeding 300K entries [4]), such systems can still suffer from redundancy and the inherent limitations of matching discrete predefined units to the continuous variability of speech.

In contrast, end-to-end deep learning models excel at capturing a wide range of complex motions [5]. However, the inherent ambiguity in the speech-to-gesture relationship (the "many-to-many mapping" problem) often leads these models to generate "average" gestures. These motions might appear fluidly human-like but frequently lack semantic alignment with the spoken content. This distinction is critical: "human-likeness" refers to the fluidity and realism of the motion itself (often high in both good rule-based and end-to-end systems, depending on data quality), while "appropriateness" relates to how well the gesture matches the specific semantic context of the speech [6]. User studies highlight this gap: end-to-end models may achieve high human-likeness scores but significantly lower appropriateness scores (e.g., 60/100) compared to ground truth gestures (75/100 for both metrics) [6]. Notably, retrieval-based systems, including many rule-based approaches, inherit the human-likeness quality directly from the underlying motion capture data.

Furthermore, capturing speaker-specific gesturing styles is vital for realism. Systems incorporating style awareness—whether through separate rule bases [7, 4], dedicated models per speaker [8], or speaker ID conditioning [2]—tend to yield superior results. Neglecting style often results in generic or averaged motions.

Low-latency generation is another critical requirement, especially for interactive applications. Rule-based methods have historically been favored in real-time systems [9, 7, 3] and continue to be employed in interactive virtual human toolkits and applications [10, 11, 12]. End-to-end generative models, often requiring sequence generation and potentially complex retargeting, can face greater latency challenges. Recent findings reinforce that larger gesture repertoires in rule-based systems improve quality [4]. Considering these factors—the need for high appropriateness and human-likeness, style preservation, and low latency—motivates the development of systems that leverage large, speaker-aware gesture libraries combined with efficient, context-sensitive retrieval or generation mechanisms.

Large Language Models (LLMs) have emerged as a powerful tool for integrating semantic understanding into gesture generation. They have been used variously to provide input embeddings [4], generate textual descriptions to guide gesture selection [13], or map speech content to specific gestures via instruction tuning [14].

To address the multifaceted challenges outlined above, we introduce **RIDGE** (Rule-Integrated Deep Gesture Engine), a hybrid system that strategically combines the strengths of rule-based precision and deep learning flexibility. RIDGE first attempts to match input text segments against an automatically constructed rule base using cosine similarity. This rule base contains key phrases extracted from transcripts using an LLM (ChatGPT) paired with corresponding high-fidelity motion segments. If the similarity score for a match exceeds a predetermined threshold, the highly relevant, prerecorded gesture is retrieved. This retrieval ensures high human-likeness, reflecting the quality of the source data. If no rule-based match meets the threshold, the system seamlessly transitions to a deep learning module to select an appropriate gesture.

The deep learning component employs a contrastive learning framework, inspired by CLIP [15] and SimCLR [16], to learn a shared latent space for text and gestures. It uses a BERT-based text encoder and a gesture encoder adapted from a pre-trained GestureCLR model [17]. Alignment between text and gesture embeddings in this space is enforced using cosine

similarity and the NT-Xent loss [16]. Crucially, the model is pre-trained on a large, diverse video dataset [17, 4] before being fine-tuned on specific, high-quality motion capture data. This strategy allows the model to learn general text-gesture semantic relationships, differentiate appropriate from inappropriate gestures (mitigating the "average gesture" problem), and adapt effectively to smaller, cleaner datasets or specific speaker styles. Further details on the architecture are in Section 3.

We evaluate RIDGE using a novel metric, Gesture Cluster Affinity (GCA), detailed in Section 5. Building on prior work showing clustering enhances gesture diversity [18, 4], GCA quantifies semantic text-gesture alignment through hierarchical clustering. RIDGE achieves a GCA score of 0.73, significantly outperforming baseline automatic rule-based ([4], score 0.60) and end-to-end systems ([19] baseline, score 0.52), and closing the gap towards the ground truth score of 0.90 observed on the BEAT dataset.

While the current implementation leverages approximately 70 hours of high-fidelity MoCap data and 300 hours of processed video data, we acknowledge the potential benefits of larger, even higher-quality datasets. Future work includes expanding data resources and further refining synthesis techniques.

In summary, RIDGE presents a robust and adaptable hybrid framework for co-speech gesture synthesis. By integrating deterministic rule-based matching for known phrases with semantically-aware deep learning for novel input, it balances precision, appropriateness, and naturalness, addressing key limitations of prior approaches. The subsequent sections detail the system architecture (Section 3), data preparation (Section 4), evaluation methodology (Section 5), and discuss results and future directions (Sections 6 and 7).

2 Related Work

2.1 Historical Overview

Early work in co-speech gesture synthesis primarily relied on rule-based approaches. Seminal systems such as the Animated Conversation [20] and the BEAT toolkit [9] used handcrafted rules, informed by linguistic theories (e.g., iconic, deictic, beat gestures), to generate synchronized non-verbal behavior. While these approaches demonstrated the feasibility of integrating gestures with speech, their rigid, domain-specific rule sets often led to repetitive outputs and limited interactivity.

2.2 Recent Advances in Gesture Synthesis

In contrast, recent advances have focused on data-driven methods. Initial deep learning approaches utilized LSTM and bidirectional RNN architectures [21] to map audio features to gestures, though they frequently produced averaged, less expressive motions. Later works, such as [22] and Transformer-based models [2], improved gesture diversity and semantic alignment. More recent methods incorporate contrastive and multimodal learning [23, 24] to better align speech with gesture representations, thereby yielding more contextually appropriate gestures [25].

2.3 Hybrid Approaches and ECAs

Hybrid methods combine rule-based and deep learning strategies to leverage the strengths of both. Systems like GestureMaster [26] retrieve candidate gestures from large databases and refine them with neural networks, while emerging frameworks employ large language models to guide gesture synthesis [13]. Semantic Gesticulator [14] used LLM to choose specific semantic gestures and insert them into the long gesture sequences. A more controllable method used [3] used strong semantic mapping and weak automatic mapping to create coherent gesture sequences. Using pose sequences from videos to extend the semantic reach of 3D gestures also showed promising results[17, 4, 14]. This hybridization has also influenced the evolution of embodied conversational agents (ECAs). Early ECAs [20, 9] have evolved into modern AI-driven avatars capable of real-time, natural interactions enhanced by advanced gesture synthesis [4, 26, 14].

2.4 Evaluation

Evaluation of gesture synthesis techniques typically combines objective metrics (e.g., Fréchet Gesture Distance, Beat Consistency) with subjective user studies [27]. The GENEA Challenge [27] has provided a benchmark framework, demonstrating that hybrid approaches generally achieve higher ratings in naturalness and appropriateness compared to purely rule-based or end-to-end systems.

3 System Architecture

This section details the architecture of RIDGE, a system designed to generate synchronized co-speech gestures by integrating a rule-based module and a deep learning model. The overall pipeline is illustrated in Fig. 1. We describe the components, their interaction, and the operational flow.

3.1 Rule-Based Gesture Generation Module

The rule-based module aims to retrieve gestures directly corresponding to specific phrases identified in the input text, leveraging pre-defined text-gesture pairings. The core of this module is a rule base constructed using a large language model (LLM). The LLM processes transcripts with word-level timing (from TextGrid files) using a specific prompt designed to extract key phrases likely to align with co-speech gestures. *Task: Extract Key Gesture-Aligned Phrases. Description: Read the given content of the TextGrid file, format it into a clean and coherent paragraph. Identify and extract a few key phrases that are most likely to align well with co-speech gestures. Phrases should have a minimum length of 3 words and a maximum of 10 words. Focus on phrases that are meaningful, impactful, or highlight significant actions, emotions, or intentions. Limit the selection to phrases that stand out as the most gesture-relevant, avoiding over-extraction.* Once the LLM identifies key phrases, an algorithm matches these phrases back to the timed transcript to extract precise start and end times. The corresponding gesture sequence from the original motion capture data, aligned with this timing, is extracted. Each resulting (key phrase, gesture sequence, timing) tuple is stored in the rule base. To validate this LLM-based extraction, the same text was provided to a human annotator familiar with co-speech gestures, using identical instructions. The LLM output showed approximately 80% overlap with the human annotations (see Fig. 2 for a sample comparison). In cases where a phrase might map to multiple gestures in the original data (ambiguity), the rule base retains the specific gesture originally performed by the speaker for that instance, treating the observed pairing as the standard. During inference, incoming text is segmented into overlapping chunks of 3 to 10 words. Each segment is encoded using a Sentence-BERT model (`all-MiniLM-L6-v2`) and compared against the text phrases stored in the rule base using cosine similarity. If a segment's similarity to a rule base entry exceeds a predefined threshold (adopted from optimized values in prior work [3]), the corresponding pre-stored gesture sequence is retrieved. If no match meets the threshold, the segment is passed to the deep learning module. This rule-matching process is designed for efficiency, incurring minimal latency.

3.2 Deep Learning Gesture Generation Module

When the rule-based module does not find a sufficiently similar phrase match, the deep learning module generates a suitable gesture. This module employs a contrastive learning framework to learn a shared embedding space for text and motion. An overview is shown in Fig. 3. The model consists of two main encoders:

- **Text Encoder:** Utilizes the `all-MiniLM-L6-v2` Sentence-BERT model. Its pooled output is passed through an additional feed-forward layer to generate a latent representation of dimension $z = 10$.
- **Gesture Encoder:** Based on the GestureCLR architecture, this encoder processes motion data and is fine-tuned alongside the text encoder to project gesture sequences into the same latent space of dimension

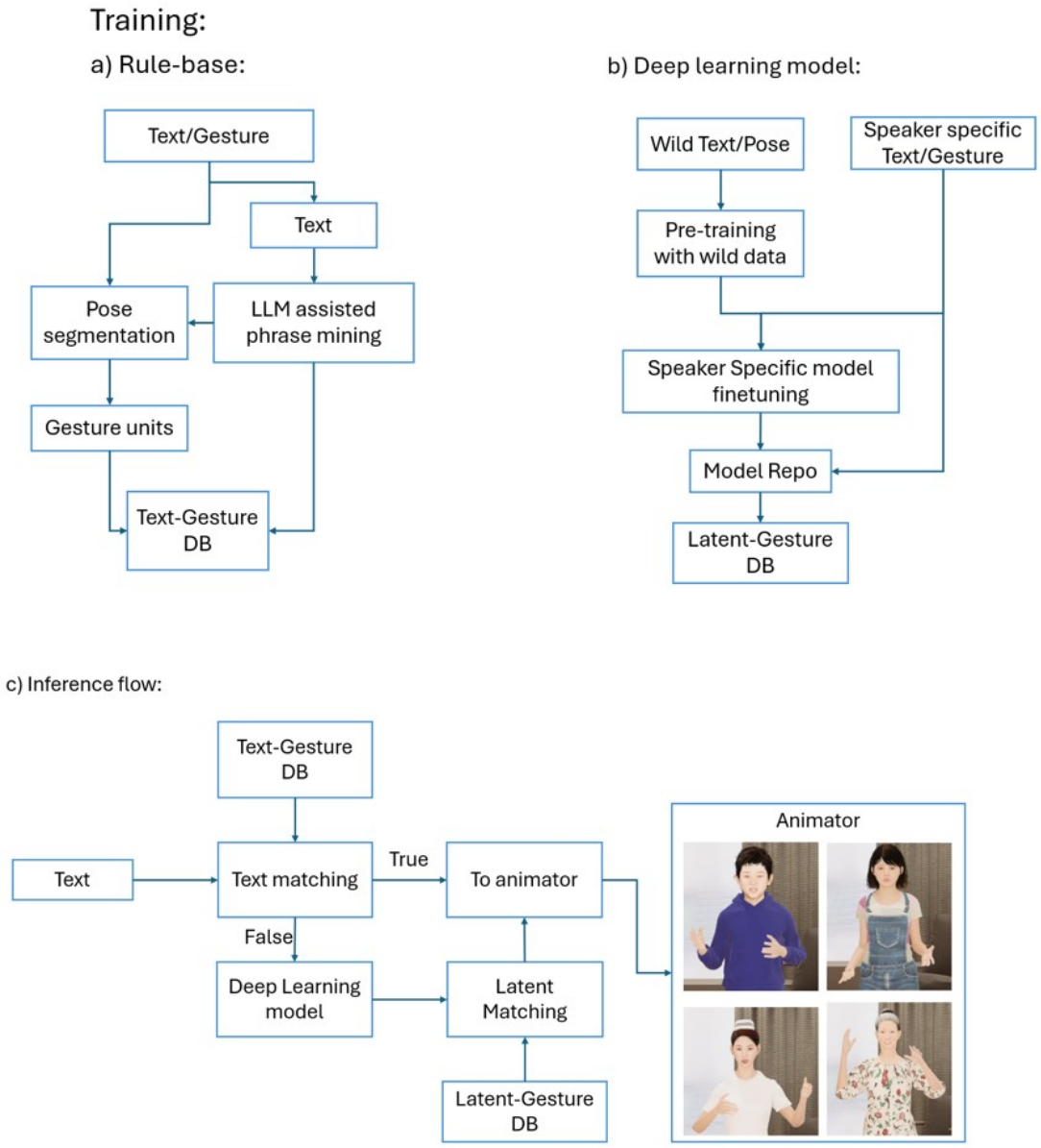

Figure 1: Overview of RIDGE pipeline. Indicating the process of rulebase and deeplearning model training

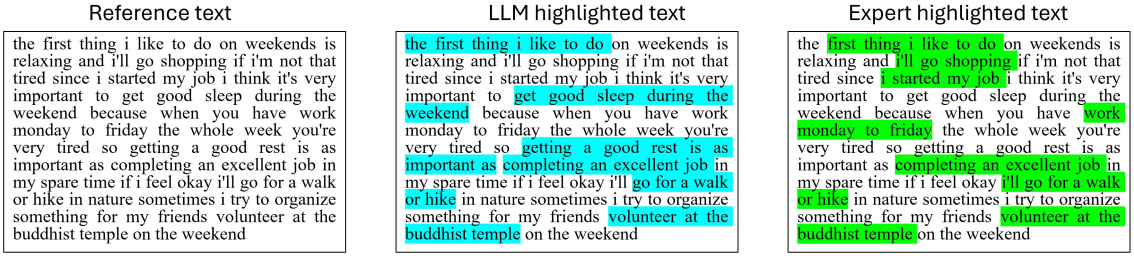

Figure 2: Sample of LLM and Human annotator's highlighted pharses.

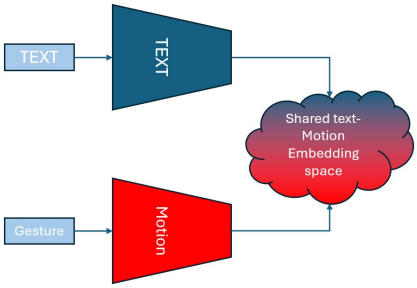

Figure 3: Overview of Deep learning model showing text and motion encoder

$z = 10$. Both encoders are trained end-to-end.

The contrastive learning objective encourages text segments and their corresponding gestures to have nearby representations in the latent space, while unrelated text-gesture pairs are pushed apart. For a given positive text-gesture pair (t, g) in a training batch, all other pairs in the batch serve as negative samples. The temperature parameter τ for the contrastive loss is tuned following methods from SimCLR [16]. To aid generalization and prevent overfitting, especially given the nature of available gesture data (often upper-body focused), the input gesture data is normalized around the neck joint before being fed into the encoder.

3.3 System Integration and Operation

The rule-based and deep learning modules operate sequentially to generate the final gesture sequence for an input text. Input text is first processed by the rule-based module's inference logic. Segments successfully matched above the similarity threshold trigger the retrieval of their corresponding animation directly from the rule base. Segments that fail to match are re-segmented (typically into 5-6 word chunks) and passed to the deep learning module. The DL module computes a latent representation for the segment. This representation is then used to query a vector database (populated with embeddings of gesture sequences from the training data) to find and retrieve the most appropriate gesture animation. The primary metric for evaluating the quality and synchronization of the generated gestures is the Gesture Cluster Affinity (GCA) score. The system is designed for real-time applications; both modules are lightweight, and the segmentation approach allows for parallel processing or streaming of input text, ensuring low latency (typically milliseconds for gesture generation). This segmentation also helps mitigate the impact of occasional mismatches, as errors are contained within small segments rather than affecting the entire sequence. Our tests indicate that gesture generation is significantly faster than upstream components like chatbots or Text-to-Speech (TTS) systems in typical interactive pipelines.

4 Data Collection and Preparation

The training and evaluation of the RIDGE system rely on two complementary datasets, providing both high-fidelity motion capture data and diverse, naturalistic gestures gathered from online videos. This section describes these datasets, their processing, and the training strategy employed.

4.1 Datasets

Two primary datasets were utilized:

- **BEAT Dataset:** This dataset [19] comprises approximately 70 hours of high-quality 3D motion capture data featuring co-speech gestures. Recorded from 20 professional actors (around 2 hours each), it provides clean, precisely captured upper-body movements synchronized with corresponding transcripts. Its high fidelity makes it ideal for fine-tuning the deep learning model and serving as a reference for data cleaning.

- **YouTube ("In the Wild") Dataset:** To incorporate a wider variety of natural gestures and speaking styles, a large dataset was compiled from publicly available online videos (e.g., TED Talks, presentations) [4]. Initial processing involved:

  – Extracting 2D pose estimations from video frames using OpenPose [28].

– Automatically transcribing the accompanying audio and aligning the text with the corresponding video frames.

4.2 Data Processing and Augmentation

Raw 2D pose data from the YouTube dataset can be noisy or incomplete. To address this and leverage the high quality of the BEAT data, we employed a data mapping technique inspired by GestureCLR[17]. A mapping model was trained using the BEAT dataset as a reference. This model learned to transform the extracted 2D poses from the YouTube videos into a cleaned, consistent 3D motion representation space, similar to that of the BEAT data. This process effectively "cleaned" the noisier YouTube gestures and allowed us to expand the usable gesture data volume significantly, resulting in approximately 300 hours of processed gesture data suitable for training. Gesture datasets often exhibit an imbalance, with common beat or metaphoric gestures being far more frequent than rarer iconic or deictic gestures. The 2D-to-3D mapping process serves as a form of data augmentation. By converting a large volume of diverse 2D gestures into a high-quality 3D space, we increase the overall number of gesture examples, including instances of rarer types. This enriched dataset helps the deep learning model learn a more robust representation of various gesture types and reduces bias towards only the most frequent motions.

4.3 Model Training Strategy

The deep learning module was trained using a two-stage strategy to benefit from both the large-scale YouTube data and the high-fidelity BEAT data.

- **Initial Training:** The model (both text and gesture encoders) was first trained on the large, processed YouTube dataset (approx. 300 hours). This phase used a batch size of 1000 and a variable learning rate, with early stopping based on validation loss. The goal was to provide the model with a robust initialization by exposing it to diverse gesture patterns.
- **Fine-Tuning:** Subsequently, the model was fine-tuned on the smaller, high-quality BEAT dataset (70 hours). A reduced batch size of 64 was used, employing the same early stopping criteria. This phase aimed to refine the model's ability to generate precise, well-synchronized gestures aligned with speech nuances captured in the MoCap data.

This pre-training/fine-tuning approach leverages the breadth of the YouTube data and the depth/quality of the BEAT data, leading to a more capable final gesture generation model.

5 Gesture Cluster Affinity (GCA)

Evaluating the semantic alignment between automatically generated gestures and their corresponding text segments is challenging. To address this, we propose the *Gesture Cluster Affinity* (GCA) metric. GCA quantifies the coherence between text and gesture segments by measuring how consistently they fall into corresponding clusters within a learned hierarchical embedding space. This section details the GCA framework, including data preparation, clustering, scoring, parameter tuning, and evaluation results.

5.1 GCA Framework and Computation

The core idea behind GCA is to first group text segments based on semantic similarity and then, within each text group, further cluster the associated gesture segments. A high GCA score indicates that text segments with similar meanings are consistently paired with gestures belonging to specific, relevant gesture sub-clusters. Figure 4 provides a conceptual overview.

Data Preparation and Embedding We use the BEAT dataset [19], containing multi-speaker co-speech gesture data. Utterances are segmented into 6-word text chunks. Corresponding gesture segments are extracted based on the timing of these text chunks. If a system being evaluated provides its own segmentation, we use that; otherwise, we apply this 6-word chunking method. For each segment pair (i), we extract embeddings:

Table 1: GCA Scores for Different Gesture Cluster Sizes (G_K) and Clustering Methods ↑

Clustering Method	$G_K = 10$	$G_K = 20$	$G_K = 30$
K-Means	0.40	0.67	0.55
Bisecting K-Means	0.49	0.75	0.52

Table 2: Comparative GCA Scores ↑

Method	**GCA Score**
RIDGE-full	0.73
RIDGE-Rulebase only	0.47
RIDGE-Model only	0.51
Multilingual Baseline	0.61
BEAT Baseline	0.52
Ground Truth	0.90

- **Text Embedding ($\mathbf{t}_i$):** Generated using a Sentence-BERT model's pooled output for the text chunk.
- **Gesture Embedding ($\mathbf{g}_i$):** Generated using a motion encoder (e.g., a contrastive learning-based model like GestureCLR) for the gesture segment.

Two-Level Clustering A hierarchical clustering process is applied:

1. **Text Clustering:** All text embeddings $\{\mathbf{t}_i\}$ are clustered into K global text clusters (e.g., $K = 100$) using k-means or bisecting k-means with cosine similarity. Let cluster$(\mathbf{t}_i) = k_i$ be the text cluster index for segment i.
2. **Gesture Sub-Clustering:** For each text cluster $k \in \{1, \ldots, K\}$, the gesture embeddings $\{\mathbf{g}_i\}$ corresponding to text segments in that cluster ($\{\mathbf{g}_i \mid k_i = k\}$) are further clustered into G_K gesture sub-clusters (e.g., $G_K \in \{10, 20, 30\}$). Let cluster$(\mathbf{g}_i \mid k_i) = m_i$ be the gesture sub-cluster index for segment i within its text cluster k_i.

GCA Score Calculation For each text-gesture segment pair $(\mathbf{t}_i, \mathbf{g}_i)$, we compute an affinity score $S(\mathbf{t}_i, \mathbf{g}_i)$. This score reflects how well the gesture $\mathbf{g}_i$ fits within the gesture sub-cluster m_i associated with its text cluster k_i. While the original definition used a binary score based on matching sub-clusters and a distance-based fallback, a common practical implementation often relies directly on the similarity of the gesture to its assigned sub-cluster centroid. Letting $\mathbf{c}_{m_i}$ be the centroid of gesture sub-cluster m_i, and using cosine similarity $\text{sim}(\cdot, \cdot)$ (where distance $d = 1 - \text{sim}$), the score can be defined proportional to this similarity. For instance, using the provided formula structure:

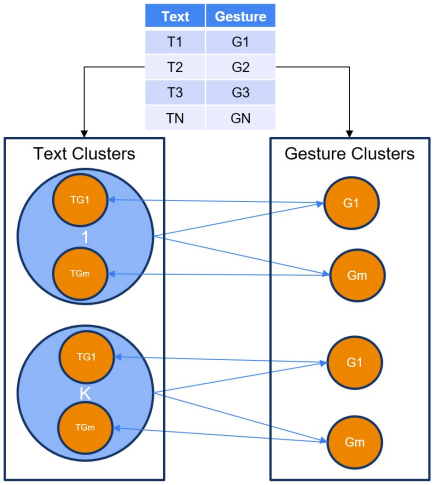

Figure 4: Overview of GCA. Text embeddings are grouped (blue circles). Within each text group, associated gestures are sub-clustered (brown circles). Brown circles in blue circles represent text sub-groupings maintaining text-gesture pairs. Lines show associations.

$$S(\mathbf{t}_i, \mathbf{g}_i) = \begin{cases} 1, & \text{if } \mathbf{g}_i \text{ perfectly matches} \\ 1 - d(\mathbf{g}_i, \mathbf{c}_{m_i}), & \text{representing similarity otherwise.} \end{cases} \quad (1)$$

The final GCA score is the average affinity across all N segments:

$$\text{GCA} = \frac{1}{N} \sum_{i=1}^{N} S(\mathbf{t}_i, \mathbf{g}_i).$$

A higher GCA score indicates better overall semantic alignment between the text and generated gestures according to the learned cluster structure.

5.2 Parameter Tuning, Evaluation, and Runtime Use

Parameter Tuning Optimal values for the number of text clusters (K) and gesture sub-clusters (G_K) are determined via cross-validation. The BEAT dataset (20 speakers) is repeatedly split into 15 speakers for building the cluster structure and 5 speakers for validation. This speaker-independent split enhances generalization assessment. The parameters (K, G_K) yielding the best average GCA score on the validation sets across multiple splits are selected. After tuning, the final clustering structure used for evaluation is built using data from all 20 speakers, which may slightly alter cluster boundaries compared to the validation splits.

Experimental Evaluation We analyzed GCA's sensitivity to the number of gesture sub-clusters (G_K) and the clustering algorithm (k-means vs. bisecting k-means), with results summarized in Table 1. Bisecting k-means often performed better, potentially due to creating more balanced clusters. Increasing G_K did not always improve the GCA score, suggesting an optimal granularity exists.

We compared our RIDGE system against baselines using the GCA metric with optimized parameters (e.g., bisecting k-means, $K = 100, G_K = 20$), as shown in Table 2. Evaluation involved sampling text segments, generating corresponding gestures using each system (RIDGE, Multilingual [4], BEAT Baseline [19] adapted for text-only input), and calculating their GCA scores against the cluster structure derived from ground truth data. Ground truth pairs (original text-gesture pairs from the dataset) establish the reference score. Ablation studies (RIDGE rule-based only, deep learning model only) were also conducted. Results indicate that RIDGE outperforms the baselines and ablation models, demonstrating better text-gesture alignment, though a gap remains compared to the ground truth, indicating room for improvement.

Runtime Confidence Check Beyond offline evaluation, GCA can function as a runtime confidence measure. During generation, if a generated gesture $\mathbf{g}_i$ has a large distance $d(\mathbf{g}_i, \mathbf{c}_{m_i})$ to its assigned sub-cluster centroid (i.e., low affinity $S(\mathbf{t}_i, \mathbf{g}_i)$), potentially falling below a predefined threshold, the system could trigger corrective actions like re-generating the gesture or using a default motion.

6 Discussion

RIDGE offers a novel approach to co-speech gesture generation by synergistically combining an LLM-driven rule base with a deep learning model. A key advantage is the automated construction of the rule base, where Large Language Models extract salient phrases linked to gestures; human validation confirmed the effectiveness of this extraction (approx. 80% overlap). This rule-based component provides reliable, precisely timed gestures for recognizable phrases. Complementing this, the deep learning module generalizes gesture generation for novel text segments, benefiting significantly from our data mapping strategy which expands a limited high-fidelity dataset (BEAT) into a larger, more diverse corpus (approx. 300 hours), mitigating common data imbalance issues.

Our evaluation leverages the proposed Gesture Cluster Affinity (GCA) metric, which proved sensitive to clustering parameters and offers potential for runtime confidence assessment. Comparative results against baseline methods are promising, indicating RIDGE's capability for generating well-aligned gestures. The system's design, utilizing discrete gesture units (small animation clips retrieved based on text), facilitates integration into real-world applications like game engines (tested in Unity3D) using pre-processed, retargeted animations. This modularity also allows for diverse

outputs for similar inputs by sampling from relevant gesture clusters [4].

While the current implementation retrieves pre-existing animation segments, the underlying framework operates in a shared latent space. This architecture opens possibilities for future work involving direct decoding of latent vectors into animation frames, akin to other generative models, although this was beyond the scope of the current study. We also acknowledge limitations related to the source animation quality; while we validated RIDGE using the open BEAT dataset [19], its animations, particularly finger details, may require further cleaning for production systems. However, the framework facilitates the integration of cleaner or custom datasets.

Regarding performance, RIDGE is computationally lightweight, capable of processing numerous segments rapidly on standard hardware, ensuring suitability for interactive scenarios. In typical pipelines involving chatbots or Text-to-Speech (TTS) systems, gesture generation is unlikely to be the primary bottleneck. Finally, ethical considerations, including potential misuse (e.g., deepfakes) and ensuring cultural appropriateness of gestures, are crucial aspects that warrant ongoing attention. Similarly, validating the GCA metric against human perceptual studies remains an important step to confirm its correlation with user-perceived quality.

7 Conclusion and Future Work

In this paper, we presented **RIDGE**, a hybrid framework for co-speech gesture generation that integrates an LLM-powered rule base with a contrastive deep learning model. RIDGE generates gestures that are semantically grounded in the accompanying text via the rule base, while the deep learning component ensures natural and diverse motion for novel inputs. We also introduced the Gesture Cluster Affinity (GCA) metric, a novel quantitative method for evaluating text-gesture semantic alignment based on cluster coherence.

Future work will proceed along several key directions. Firstly, we plan comprehensive evaluation enhancements, including comparisons against a wider range of state-of-the-art baseline systems and employing established metrics like Fréchet Gesture Distance (FGD). Crucially, user studies will be conducted to validate the GCA metric's correlation with human judgments of gesture quality and appropriateness. Secondly, we aim to extend the system's capabilities by investigating the direct decoding of gesture representations from the learned latent space into animation frames. We will also explore the integration and performance of RIDGE with higher-quality, production-ready animation datasets. Finally, we will continue to address the ethical dimensions of generated human motion, focusing on responsible development and deployment practices.

References

[1] Chaitanya Ahuja, Dong Won Lee, Ryo Ishii, and Louis-Philippe Morency. No gestures left behind: Learning relationships between spoken language and freeform gestures. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1884–1895, 2020.

[2] Uttaran Bhattacharya, Nicholas Rewkowski, Abhishek Banerjee, Pooja Guhan, Aniket Bera, and Dinesh Manocha. Text2gestures: A transformer-based network for generating emotive body gestures for virtual agents. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pages 1–10. IEEE, 2021.

[3] Ghazanfar Ali, Myungho. Lee, and Jae-In Hwang. Automatic text-to-gesture rule generation for embodied conversational agents. *Computer Animation and Virtual Worlds*, 31(4-5):e1944, 2020.

[4] Ghazanfar Ali, Woojoo Kim, Shahid Anwar, Muhammad, Jae-In Hwang, and Ahyoung Choi. Expanding multilingual co-speech interaction: The impact of enhanced gesture units in text-to-gesture synthesis for digital humans. sep 2023. Preprint: https://doi.org/10.21203/rs.3.rs-3350470/v1.

[5] Simbarashe Nyatsanga, Taras Kucherenko, Chaitanya Ahuja, Gustav Eje Henter, and

Michael Neff. A comprehensive review of data-driven co-speech gesture generation. 42, 1 2023.

[6] Youngwoo Yoon, Pieter Wolfert, Taras Kucherenko, Carla Viegas, Teodor Nikolov, Mihail Tsakov, and Gustav Eje Henter. The genea challenge 2022: A large evaluation of data-driven co-speech gesture generation. *ACM International Conference Proceeding Series*, pages 736–747, 11 2022.

[7] Jina Lee and Stacy Marsella. Nonverbal behavior generator for embodied conversational agents. In Jonathan Gratch, Michael Young, Ruth Aylett, Daniel Ballin, and Patrick Olivier, editors, *Intelligent Virtual Agents*, pages 243–255, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[8] Shiry Ginosar, Amir Bar, Gefen Kohavi, Caroline Chan, Andrew Owens, and Jitendra Malik. Learning individual styles of conversational gesture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3497–3506, 2019.

[9] Justine Cassell, Hannes Vilhjálmsson, and Timothy Bickmore. Beat: The behavior expression animation toolkit. In H. Prendinger and M. Ishizuka, editors, *Life-Like Characters: Tools, Affective Functions, and Applications*, pages 163–185. Springer, 2004.

[10] VHTOOLKIT — Homepage — vhtoolkit.ict.usc.edu. `https://vhtoolkit.ict.usc.edu/`.

[11] Hanseob Kim, Ghazanfar Ali, Bin Han, Hwang Youn Kim, Jieun Kim, Hyemin Shin, Gerard Jounghyun Kim, and Jae-In Hwang. Asap for multi-outputs: auto-generating storyboard and previsualization with virtual actors based on screenplay. *Multimedia Tools and Applications*, pages 1–24, 2024.

[12] Hwang Youn Kim, Ghazanfar Ali, and Jae-In Hwang. Enhancing doctor-patient communication in surgical explanations: Designing effective facial expressions and gestures for animated physician characters. *Computer Animation and Virtual Worlds*, 35(3):e2236, 2024.

[13] Haozhou Pang, Tianwei Ding, Lanshan He, Ming Tao, Lu Zhang, and Qi Gan. LLM gesticulator: Leveraging large language models for scalable and controllable co-speech gesture synthesis. arXiv preprint arXiv:2410.10851, 2024.

[14] Zeyi Zhang, Tenglong Ao, Yuyao Zhang, Qingzhe Gao, Chuan Lin, Baoquan Chen, and Libin Liu. Semantic gesticulator: Semantics-aware co-speech gesture synthesis. *ACM Trans. Graph.*

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[16] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[17] Ghazanfar Ali and Jae-In Hwang. Improving co-speech gesture rule-map generation via wild pose matching with gesture units. *SIGGRAPH Asia 2022 Posters*, PartF168147-3:1575–1585, 12 2022.

[18] Carolyn Saund, Andrei Birladeanu, and Stacy Marsella. Cmcf: An architecture for realtime gesture generation by clustering gestures by motion and communicative function. 2021.

[19] Haiyang Liu, Zihao Zhu, Naoya Iwamoto, Yichen Peng, Zhengqing Li, You Zhou, Elif Bozkurt, and Bo Zheng. Beat: A large-scale semantic and emotional multimodal dataset for conversational gestures synthesis. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 13667 LNCS:612–630, 2022.

[20] Justine Cassell, Catherine Pelachaud, Norman Badler, Mark Steedman, Brett Achorn, Badri Ambacheri, Tripp Becker, Brett Douville, Scott Prevost, and Matthew Stone. Animated conversation: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In *Proceedings of SIGGRAPH '94 (21st Annual Conference on Computer Graphics and Interactive Techniques)*, pages 413–420. ACM, 1994.

[21] Dai Hasegawa, Naoshi Kaneko, Shinichi Shirakawa, Hiroshi Sakuta, and Kazuhiko Sumi. Evaluation of speech-to-gesture generation using bi-directional LSTM network. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents (IVA)*, pages 79–86. ACM, 2018.

[22] Jing Li, Di Kang, Wenjie Pei, Xuefei Zhe, Ying Zhang, Zhenyu He, and Linchao Bao. Audio2Gestures: Generating diverse gestures from speech audio with conditional VAE. arXiv preprint arXiv:2108.06720, 2021.

[23] Xian Liu, Qianyi Wu, Hang Zhou, Yinghao Xu, Rui Qian, Xinyi Lin, Xiaowei Zhou, Wayne Wu, Bo Dai, and Bolei Zhou. Learning hierarchical cross-modal association for co-speech gesture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11478–11488, 2022.

[24] Mingyang Sun, Mengchen Zhao, Yaqing Hou, Minglei Li, Huang Xu, Songcen Xu, and Jianye Hao. Co-speech gesture synthesis by reinforcement learning with contrastive pre-trained rewards. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2321–2331, 2023.

[25] Yoonwoo Yoon, Bok Choy, Jinhyeong Kim, Jaeyeon Lee, Minju Jang, Jaegyun Lee, and Gyeongyu Lee. Speech gesture generation from the trimodal context of text, audio, and speaker identity. *ACM Transactions on Graphics*, 39(6):219:1–219:16, 2020.

[26] Chen Zhou, Tianyuan Bian, and Kang Chen. Gesturemaster: Graph-based speech-driven gesture generation. In *Proceedings of the International Conference on Multimodal Interaction (ICMI)*. ACM, 2022.

[27] Peter Wolfert, Jules M. G. Girard, Taras Kucherenko, and Tony Belpaeme. Investigating evaluation methods for generated co-speech gestures. In *Proceedings of the 2021 International Conference on Multimodal Interaction (ICMI)*, pages 205–214. ACM, 2021.

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

8 Author Biography

Ghazanfar Ali completed his Ph.D. from UST in 2023. His research topic primarily focuses on infusing emotional Intelligence into virtual humans in AR/MR scenarios by leveraging deep learning of human behavior. He did his B.E Computer Engineering from NUST – Pakistan in 2014. He is a former Faculty Member at a Pakistani University. In the Fall of 2017, he came to KIST as a UST student in the Integrative program and started working in the MR Lab.

Hwang Youn Kim is a Ph.D. student at AI-Robotics, University of Science and Technology. He completed a Master of Entertainment Technology at Carnegie Mellon University in 2015. He had work experience

as a developer in various fields, such as educational games, avatar control systems, and healthcare applications. Now, he is interested in the integration of virtual humans and deep learning models.

Jae-In Hwang is a professor at KIST School and a principal research scientist in Center for Artificial Intelligence at Korea Institute of Science and Technology. He received the B.S., M.S., and Ph.D. degrees in computer science and engineering at Pohang University of Science and Technology, Korea, in 1998, 2000, and 2007, respectively.