Supplementary Material

This document provides additional analysis and details on the proposed hybrid garment animation framework, addressing specific points raised by reviewers.

0.1 Motion Similarity vs. Error Analysis

To validate the robustness of our proposed framework across different motion distributions, we analyzed the relationship between input motion similarity to the training database and the model's prediction error. This analysis addresses concerns about the method's performance on out-of-distribution motions.

0.1.1 Methodology

Motion similarity is computed as the average Euclidean distance between the normalized input motion features and the top-10 most similar motions in the database. Model performance is measured using Mean Vertex Error (cm). Lower similarity distances indicate that the input motion lies within the training distribution, while higher distances suggest out-of-distribution or unseen motions.

0.1.2 Results and Analysis

Table 1 presents the relationship between motion similarity and prediction error across different similarity distance ranges.

Table 1: Motion Similarity vs. Prediction Error Analysis

Similarity Distance (Top-10 Avg.)	Mean Vertex Error (cm)	Std Dev (cm)	Median (cm)	IQR (cm)
0.29	0.59	0.39	0.50	0.38
0.34	0.61	0.36	0.52	0.41
0.40	0.58	0.34	0.51	0.39
0.45	0.56	0.33	0.50	0.37
0.51	0.57	0.32	0.49	0.36
0.56	0.61	0.40	0.51	0.42
0.62	0.67	0.36	0.60	0.45
0.67	0.71	0.31	0.66	0.29

Key Observations:

- **Stable Performance Across Motion Distributions.** The model maintains consistent error levels regardless of motion similarity to the training database, demonstrating the effectiveness of our hybrid framework's regression fallback mechanism.
- **Non-monotonic Relationship.** Interestingly, higher motion similarity does not always guarantee lower prediction error. This phenomenon can be attributed to the challenge of distinguishing between highly similar motion clips (e.g., frames 1-10 vs. frames 2-11) that have minimal semantic differences but may confuse the model's latent space representation.
- **Robustness to Out-of-Distribution Motions.** Even for motions with high similarity distances (indicating out-of-distribution inputs), the model maintains reasonable performance through the regression pathway, validating the hybrid design's effectiveness.

This analysis empirically demonstrates that our model does not over-rely on retrieval-based matching and maintains robust prediction performance across the entire motion distribution spectrum.

0.2 Temporal Coherence Analysis

0.2.1 Stride Configuration Impact

Our framework processes motion input using sliding windows with configurable stride values. We analyze the trade-off between temporal coherence, accuracy, and computational efficiency across different stride settings.

0.2.2 Methodology

Temporal coherence is quantitatively evaluated using the variance of the frame-wise vertex mean error. We suppress jittering artifacts at window boundaries through averaging overlapping regions, which is enabled by a stride of 1. Different stride values are evaluated to understand their impact on temporal smoothness and computational cost.

0.2.3 Results

Table 2 presents the performance comparison across different stride configurations.

Table 2: Stride Configuration Impact on Performance

Stride	Mean Vertex Error (mm)	Std (mm)	Execution Time per Frame (ms)
1	6.5 ± 3.3	5.74	56
4	6.9 ± 3.5	2.92	10
7	7.5 ± 4.2	2.37	7

Analysis:

- **Stride=1** provides the best accuracy and temporal smoothness through maximum overlap averaging but requires a higher computational cost.
- **Larger strides** reduce computation time but risk temporal discontinuities and accuracy degradation.

The chosen stride=1 configuration optimally balances quality and practical performance requirements.

0.3 Model Implementation Details

Our hybrid garment animation framework consists of four main components: a pre-trained mesh autoencoder, a motion encoder with a spatio-temporal attention estimator, a garment VQ decoder, and a Gumbel-Softmax-based vector quantization mechanism. This section provides the full implementation details of our architecture and training setup.

- **Motion Encoder Architecture.** The motion encoder is designed to map a sequence of motion features into a shared discrete latent space. It is a GRU-based architecture with a temporal attention refinement module. For each frame, the input is a feature vector $\mathbf{f} \in R^{124}$ capturing joint posture, velocity, and global body dynamics. A GRU layer processes the sequence of these features to capture temporal context. The output of the GRU is then refined by a multi-head temporal attention mechanism with a causal mask, ensuring that predictions at each time step only depend on past and current frames. This attention module enhances the model's ability to learn intricate temporal dependencies crucial for loose-fitting garment dynamics. The refined output is then passed through a fully connected layer to produce the codebook logits.
- **Garment VQ Encoder Architecture.** This encoder maps a sequence of garment latent vectors, extracted from a pre-trained mesh autoencoder, into the same shared discrete latent space as the motion encoder. It consists of a three-layer fully connected network with ELU activations and dropout regularization. The input is a sequence of garment latents, which are processed to output logits of a size equal to the codebook size.

- **Gumbel-Softmax Vector Quantization.** We employ Gumbel-Softmax-based vector quantization to enable end-to-end differentiable training with discrete representations. During training, Gumbel noise is added to the logits, followed by a temperature-scaled softmax to get soft assignments. We use the straight-through estimator (STE) to approximate gradients, allowing backpropagation through the discrete bottleneck. For inference, noise injection is disabled, and a deterministic hard assignment (argmax) is used to select a single codebook entry.

- **Garment VQ Decoder Architecture.** The decoder reconstructs garment latent sequences from the quantized discrete codes. It is a symmetric three-layer fully connected network with ELU activations, taking the quantized codebook vectors as input. It progressively transforms and expands the feature dimensionality to output the predicted garment latent sequence. The final garment mesh is then reconstructed from these latents using the decoder of the pre-trained mesh autoencoder.

- **Training Configuration.** Our model is trained using a two-stage curriculum learning approach. Initially, the reconstruction loss ($\mathcal{L}_{rec}$) is weighted more heavily ($\lambda_{rec} = 1.0$, $\lambda_{match} = 0.1$) to stabilize the latent space. Once the reconstruction loss plateaus, the code matching loss ($\mathcal{L}_{match}$) weight is increased ($\lambda_{match} = 0.5$) to emphasize cross-modal alignment. The Adam optimizer is used with a learning rate of 1e-4 and a weight decay of 1e-4 for 1500 epochs.

- **Inference Strategy.** During inference, the motion encoder produces logits for the codebook. We compute the entropy of the resulting softmax distribution to measure confidence. If the entropy is below an empirically determined threshold $\tau = 0.5$, a high-fidelity garment sequence is retrieved from a precomputed database. Otherwise, if the confidence is low, the motion code is fed to the VQ decoder to regress the garment sequence. This dynamic switching enables robust handling of both familiar and ambiguous motion patterns.

0.4 Limitations and Future Directions

0.4.1 Current Limitations

- **Fixed Body and Garment Assumption.** Our current framework assumes fixed body shapes and specific garment types during training and inference.

- **Database Storage Requirement.** The retrieval component requires maintaining a database of motion-garment pairs, which increases the memory footprint.

0.4.2 Future Research Directions

- **Garment-Agnostic Learning.** Extending the framework to learn garment-type-agnostic latent spaces could enable cross-garment generalization and broader applicability.

- **Database-Free Architecture.** Inspired by learned motion matching techniques, developing approaches that eliminate the need for explicit database storage while maintaining performance could improve practical deployment efficiency.

- **Contrastive Learning Enhancement.** Implementing contrastive learning strategies could improve latent space quality by better distinguishing between similar motion clips, addressing the similarity-error relationship observations.

These directions represent promising avenues for enhancing the framework's generalization capabilities and practical applicability in real-world scenarios.