

SSLR: Saudi Sign Language Recognizer

Alaqsa Akbar¹, Marwan Alkhayyat¹, and Mohammad Jaddoua¹

¹King Fahd University of Petroleum and Minerals

Abstract

Sign language serves as a crucial communication medium for the deaf and hard-of-hearing community, yet automated recognition systems for Arabic Sign Language remain under-explored. In this paper, we present SSLR (Saudi Sign Language Recognizer), a robust pose-based recognition system. Our architecture leverages a two-stream approach utilizing Grouped Graph Convolutional Networks (GCNs) for spatial feature extraction and Transformers for temporal modeling. We conduct a comprehensive comparative analysis of various training strategies, including Masked Autoencoder (MAE) and Contrastive pretraining, against a Hybrid Loss objective combining Connectionist Temporal Classification (CTC) and Cross Entropy (CE). Our experimental results demonstrate that the Hybrid Loss strategy significantly outperforms pretraining methods, achieving a Word Error Rate (WER) of 6.74% and an accuracy of 89.67%. While contrastive pretraining exhibited faster convergence, we identify challenges in fine-tuning pretrained models with CTC-based objectives. We open-source our code and models to facilitate further research in Arabic SLR.

1 Introduction

Sign languages are complex, structured visual languages that serve as the primary means of communication for the deaf and hard-of-hearing community. Much like spoken languages, sign languages are not universal; they possess distinct grammatical structures, vocabularies, and regional variations that render them mutually unintelligible across different geographies. Despite the World Health Organization estimating that disabling hearing loss affects over 430 million people globally, technological accommodation for sign language remains significantly underdeveloped compared to spoken languages. This creates a communication barrier that limits accessibility to essential services and information. Consequently, developing robust, high-quality Sign Language Recognition (SLR) systems is of paramount importance to bridge this digital divide.

Sign Language Recognition (SLR) aims to translate continuous video streams or pose sequences into a structured textual format. A critical component of this process is the detection of "glosses"—an intermediate textual representation

where each gloss maps one-to-one with a specific sign gesture, often disregarding the syntax of the target spoken language. For instance, in American Sign Language (ASL), the English sentence “Today is Friday” is represented by the gloss sequence “NOW DAY FRIDAY.” In this context, the challenge lies in accurately mapping spatiotemporal pose sequences to these discrete gloss units.

In this paper, we present **SSLR** (Saudi Sign Language Recognizer), a novel framework designed to achieve state-of-the-art results on Arabic Sign Language tasks. We propose a robust two-stream architecture that leverages Grouped Graph Convolutional Networks (GCNs) for spatial feature extraction and Transformers for temporal modeling. Furthermore, we conduct a rigorous comparative analysis of training strategies, specifically evaluating the efficacy of self-supervised methods—such as Masked Autoencoders (MAE) and Contrastive Learning—against a supervised Hybrid Loss (CTC and Cross Entropy) approach. To facilitate further research in this domain, we make our implementation publicly available at: <https://github.com/alaqsa-akbar/SSLR>.

2 Background

2.1 Graph Convolutional Networks (GCNs)

Graph Convolutional Networks (GCNs) are specialized neural networks designed to operate on graph-structured data. In the context of SLR, the human body is modeled as a graph where joints (keypoints) represent nodes and bones represent edges. GCNs exploit the natural topology of the skeleton, allowing the model to learn spatial dependencies between adjacent joints (e.g., the relationship between the wrist and the elbow). Unlike standard Convolutional Neural Networks (CNNs) that operate on a fixed grid, GCNs can effectively capture the irregular geometric structure of human pose data, making them ideal for extracting spatial features from sign language gestures.

2.2 Transformers and Cross Entropy Loss

Transformers have become the state-of-the-art architecture for sequence modeling tasks due to their self-attention mechanism, which captures long-range dependencies better than Recurrent Neural Networks (RNNs). In SLR, the Transformer acts as a sequence-to-sequence model. The encoder processes the stream of visual features extracted from the frames, while the decoder generates the corresponding sequence of glosses autoregressively.

The standard objective function used for training the decoder is the Cross Entropy (CE) loss. CE measures the divergence between the predicted probability distribution over the vocabulary and the true target distribution (the ground truth gloss) at each time step. It forces the model to maximize the likelihood of the correct gloss sequence given the visual input.

2.3 Connectionist Temporal Classification (CTC) Loss

One of the primary challenges in SLR is the lack of frame-level alignment; the input video usually has far more frames than the output gloss sentence, and the exact boundaries of each sign are unknown. Connectionist Temporal Classification (CTC) loss addresses this by allowing the network to output a probability distribution over all possible labels (plus a special "blank" token) for each frame. It then sums over all possible alignments of the input frames to the target sequence via dynamic programming. This enables the model to learn temporal alignments without explicit boundary supervision.

2.4 Contrastive Training

Contrastive representation learning aims to learn an embedding space where semantically similar samples are pulled closer together while dissimilar samples are pushed apart. In the context of multi-modal SLR (Text and Pose), we utilize an InfoNCE-style loss. This approach encourages the pose encoder to produce embeddings that are aligned with the embeddings of the corresponding text glosses (generated by a pre-trained language model) while differentiating them from unrelated glosses. This alignment aids the model in learning robust, semantic-rich visual representations before the decoding stage.

3 Methodology

3.1 Data Preprocessing

To enhance the generalization capabilities of the model, we apply aggressive data augmentation techniques during training. We augment the data by injecting random noise, applying random scaling to the pose size, and performing random spatial shifts. Additionally, we randomly rotate the keypoints, modify the speed of the pose sequence (temporal scaling), and randomly drop frames to simulate varying frame rates. Finally, we normalize the pose keypoints and extract a secondary velocity stream by calculating the difference between adjacent frames, providing the model with dynamic motion features.

3.2 Architecture

Our overall system follows an encoder-decoder structure, as illustrated in Figure 1. The **Pose Encoder** compresses the input pose sequence into a sequence of high-level feature representations. These features are then passed to the **Transformer Decoder**, which autoregressively generates the output gloss sequence.

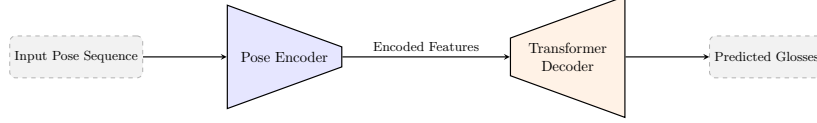


Figure 1: High-level overview of the SSLR model architecture. The Pose Encoder (narrowing trapezoid) compresses the input into latent features, which are then unfolded by the Transformer Decoder (widening trapezoid) to predict glosses.

The internal components of the **Pose Encoder** are detailed in Figure 2. It employs a two-stream approach, processing raw pose coordinates and velocity information in parallel. The keypoints are grouped (hands, body, face) and processed by dedicated GCNs to extract spatial features. These features are fused and then processed by a stack of Transformer Encoder layers to capture temporal dependencies.

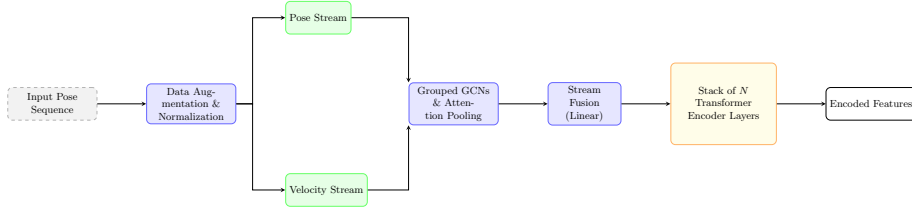


Figure 2: Detailed architecture of the Pose Encoder. It features a dual-stream GCN front-end followed by a stack of Transformer Encoder layers to process spatiotemporal information.

Figure 3 illustrates the internal structure of the Transformer Encoder and Decoder layers used in our model. The encoder layer consists of Multi-Head Attention (MHA) and a Feed-Forward Network (FFN), both with residual connections and normalization. The decoder layer adds a third sub-layer, which performs multi-head attention over the output of the encoder stack. We utilize Rotary Positional Embeddings (RoPE) and Root Mean Square Normalization (RMSNorm) for stable training.

We also train a basic transformer using PyTorch’s `nn.Transformer` as a baseline.

3.3 Training Strategies

We explore several training paradigms to optimize performance.

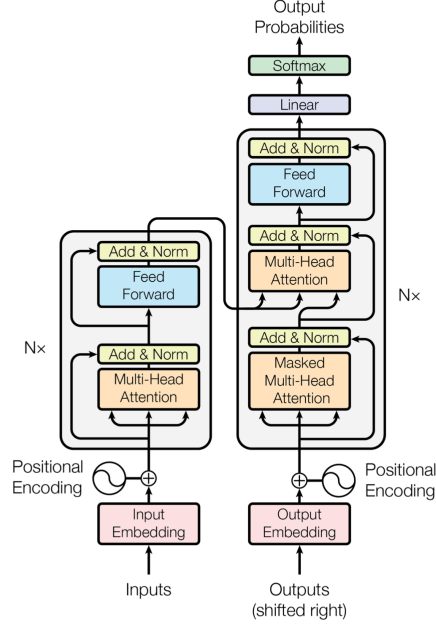


Figure 3: Internal structure of a single Transformer Encoder Layer (left) and Decoder Layer (right).

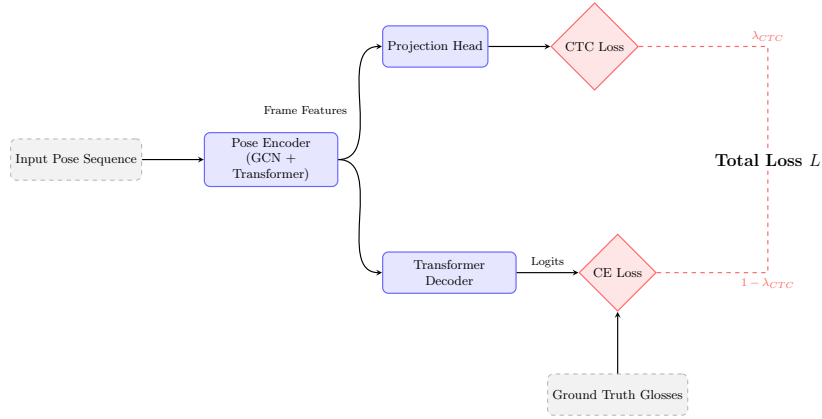


Figure 4: Block diagram of the Hybrid Loss architecture. The Pose Encoder is simultaneously supervised by the CTC Loss (for temporal alignment) and the Decoder’s Cross Entropy Loss (for sequence generation).

Hybrid Loss: We utilize a hybrid objective function combining CTC and Cross Entropy (CE) to supervise the model, as illustrated in Figure 4. The total

loss L is calculated as:

$$L = \lambda_{CTC} \times \mathcal{L}_{CTC} + (1 - \lambda_{CTC}) \times \mathcal{L}_{CE} \quad (1)$$

The CTC loss is applied to the output of the pose encoder via a projection head, forcing the encoder to learn temporal alignment. The CE loss is applied to the output of the decoder. This dual-supervision ensures the encoder learns robust alignment features while the decoder focuses on language modeling.

Pretraining: We investigate two pretraining strategies:

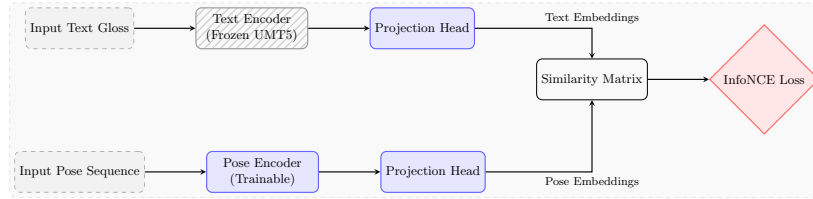


Figure 5: Architecture for Contrastive Pretraining. Visual features from the Pose Encoder are aligned with semantic text embeddings generated by a frozen pre-trained language model using InfoNCE loss.

1. **Contrastive Learning (Figure 5):** We utilize a UMT5-base encoder to generate text embeddings. The pose encoder is trained using an InfoNCE loss to align visual features with these text embeddings. A linear projection head is applied to map visual features into the shared embedding space. To maximize the number of negative samples, we implement a caching mechanism.

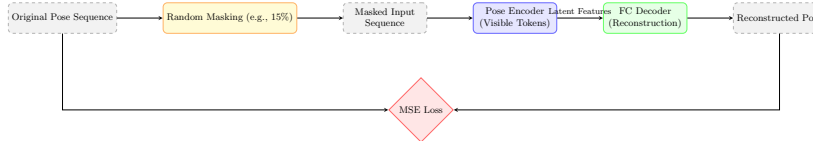


Figure 6: Masked Autoencoder (MAE) pretraining architecture. The model learns to reconstruct missing parts of the pose sequence, forcing it to learn spatiotemporal dependencies.

2. **Masked Autoencoder (MAE) (Figure 6):** The pose encoder is trained to reconstruct masked sections of the input pose sequence using a Fully Connected (FC) decoder.

For both methods, the pose encoder is initially frozen for a set number of epochs during the fine-tuning stage to prevent catastrophic forgetting.

4 Experimental Setup

All models were trained on a single NVIDIA A100 GPU with a batch size of 64. We utilized the AdamW optimizer with default parameters ($\beta_1 = 0.9, \beta_2 = 0.999$) and a learning rate of 5×10^{-5} across all tasks. We also used GELU as the activation function throughout the model. A linear learning rate decay was employed following a linear warmup for the first 15% of training steps. For regularization, we applied label smoothing of 0.1 to the Cross Entropy loss. All tasks incorporated early stopping with a patience of 20 epochs. When using pretrained encoders, the encoder weights were frozen for the first 60 epochs.

The model architecture was configured with the following hyperparameters:

- **Input:** 86 keypoints with 2 channels (x, y).
- **Encoder/Decoder:** 1024 hidden dimension, 8 layers, 16 attention heads, and a dropout rate of 0.3.
- **GCN:** 2 layers with a hidden dimension of 256.
- **Vocabulary:** 687 classes.

For the hybrid loss experiments, we set $\lambda_{CTC} = 0.3$.

5 Results

5.1 Pre-Training Evaluation

Masked Autoencoder (MAE): The MAE approach achieved high-fidelity reconstruction of the input poses. The reconstruction error metrics on the validation set are summarized below:

- **Overall MSE:** 0.0052
- **Overall MAE:** 0.0385

Part-specific Mean Squared Error (MSE) analysis shows that the face (MSE: 0.0006) was reconstructed with the highest accuracy, followed by the body, right hand, and left hand. A visual example of the reconstruction is provided in Figure 7.

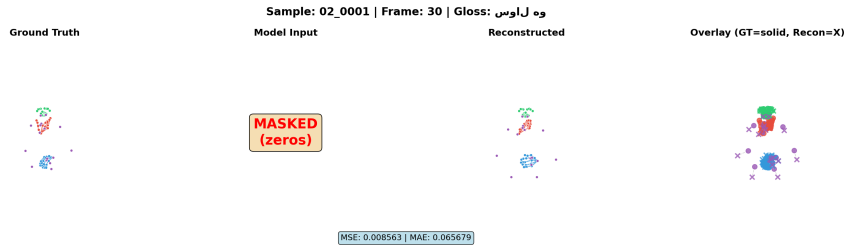


Figure 7: Visualization of MAE pose reconstruction results. The model successfully recovers masked keypoints with low error.

The training loss progression for the MAE pretraining is shown in Figure 8.

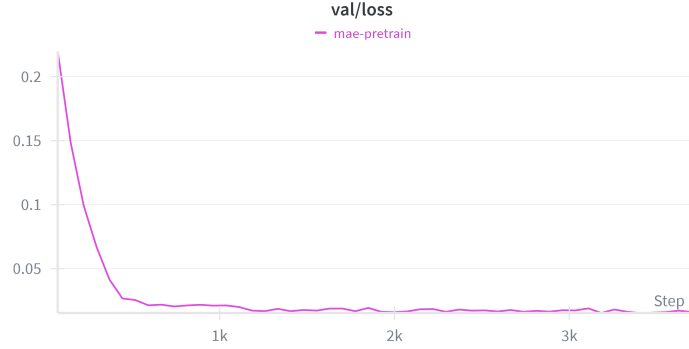


Figure 8: Training loss curve for MAE pretraining.

Contrastive Learning: The contrastive model demonstrated rapid convergence during the pretraining phase. By aligning the pose embeddings with the UMT5 text embeddings, the model learned a semantic structure prior to the decoding task. On the contrastive alignment task itself, the model achieved an accuracy of **89.01%** on the validation set. The loss curve for this pretraining phase is depicted in Figure 9.

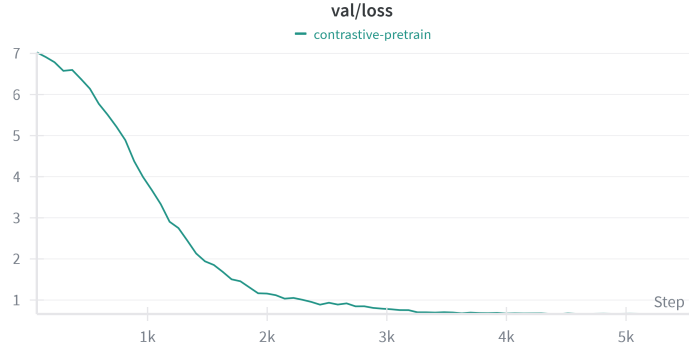


Figure 9: Training loss curve for Contrastive Learning pretraining.

5.2 Recognition Performance

We evaluated the performance of the various approaches using Word Error Rate (WER), Accuracy, BLEU-1, and BLEU-4. The results are presented in Table 1.

Table 1: Comparison of Model Performance across different training strategies.

Method	WER ($\downarrow$)	Accuracy ($\uparrow$)	BLEU-1 ($\uparrow$)	BLEU-4 ($\uparrow$)
Transformer (Baseline)	—	75.24%	—	—
No Pretrain	8.81%	88.30%	93.15	91.46
Pretrain MAE	9.35%	88.72%	92.27	90.36
Pretrain Contrastive	8.28%	88.72%	93.63	92.30
No Pretrain + CTC ($\lambda = 0.3$)	6.74%	89.67%	94.92	93.27

5.3 Analysis

Contrary to expectations, pretraining did not yield substantial benefits for this specific task. While the Contrastive Pretraining slightly improved WER (8.28%) compared to the baseline (8.81%), the MAE approach resulted in a degradation of WER (9.35%). However, the loss curve analysis (Figure 10) indicates that contrastive pretraining allows the loss to drop significantly faster than random initialization, converging to similar final results. This suggests that contrastive pretraining may be highly beneficial in data-limited scenarios where training from scratch is infeasible.

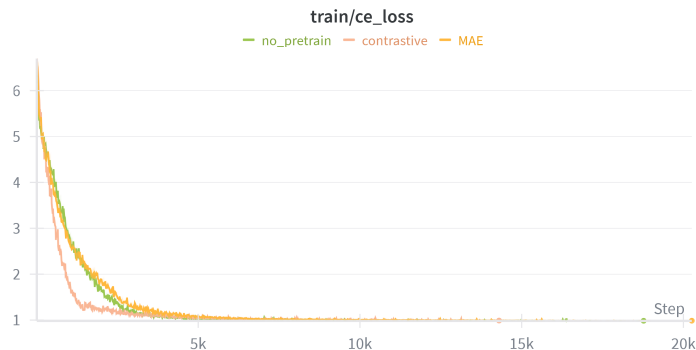


Figure 10: Training loss comparison between Pretrained and Non-Pretrained models. The contrastive pretrained model exhibits faster convergence in early epochs.

The **Hybrid Loss** approach (CTC + CE) significantly outperformed all other methods, achieving the lowest WER of 6.74%. This confirms that enforcing frame-level alignment constraints via CTC on the encoder helps the model learn more robust features for the decoder. We experimented with higher λ_{CTC} values (0.5 and 0.7), but observed that overpowering the pose encoder with CTC weight degraded performance.

Attempts to combine pretraining (MAE/Contrastive) with the Hybrid Loss resulted in performance regression. For instance, the combination of Contrastive pretraining and Hybrid Loss yielded a WER of 9.18%. We hypothesize that the

strong gradients from the CTC loss during fine-tuning heavily modify the pre-trained encoder weights, leading to catastrophic forgetting of the pretrained features. A potential resolution for future work would be to apply a lower differential learning rate to the pose encoder to preserve the pretrained representations.

6 Conclusion and Future Work

In this work, we developed and evaluated a high-performance Saudi Sign Language Recognition system. By systematically exploring architectural components and training paradigms, we demonstrated the efficacy of a GCN-Transformer hybrid architecture processed through dual pose and velocity streams. Our rigorous evaluation of training strategies revealed that while self-supervised pretraining (MAE and Contrastive Learning) accelerates convergence, a supervised Hybrid Loss (CTC + CE) yields the most robust performance, achieving a state-of-the-art WER of 6.74%. The Hybrid Loss effectively addresses the frame-level alignment problem inherent in continuous sign language recognition, proving superior to pretraining alone for this task.

Future work will focus on reconciling the benefits of pretraining with the alignment capabilities of the Hybrid Loss. Specifically, we aim to investigate differential learning rates and gradual unfreezing schedules to mitigate the catastrophic forgetting observed when fine-tuning pretrained encoders with strong CTC gradients. We would like to experiment with further hyperparameter tuning as well.