Towards Trustworthy Sign Language Translation System: A Privacy-Preserving Edge–Cloud–Blockchain Approach

Nada Shahin and Leila Ismail *

Posted Date: 2 October 2025

doi: 10.20944/preprints202510.0130.v1

Keywords: artificial intelligence; assistive technology; blockchain; cloud computing; computer vision; deep learning; edge computing; natural language processing; neural machine translation; sign language translation; transformers

 doi:10.20944/preprints202510.0130.v1

Article

Towards Trustworthy Sign Language Translation System: A Privacy-Preserving Edge–Cloud–Blockchain Approach

Nada Shahin [1,2] and Leila Ismail [1,2]*

[1] Intelligent Distributed Computing and Systems (INDUCE) Lab, Department of Computer Science and Software Engineering, College of Information Technology, United Arab Emirates University, Al Ain, Abu Dhabi, United Arab Emirates

[2] Emirates Center for Mobility Research, United Arab Emirates University, Al Ain, Abu Dhabi, United Arab Emirates

* Correspondence: leila@uaeu.ac.ae

Abstract

The growing Deaf and Hard-of-Hearing (DHH) community faces communication challenges due to a global shortage of certified sign language interpreters. Therefore, developing efficient and secure Sign Language Machine Translation (SLMT) systems is essential. This paper proposes a novel privacy-preserving end-to-end edge–cloud–blockchain system for SLMT that ensures real-time translation and enforces user consent management through immutable blockchain records. We evaluate our system by comparing the Encoder-Decoder Transformer, the mostly used model in SLMT, and our proposed Adaptive Transformer (ADAT) model. We evaluate the system on two datasets: RWTH-PHOENIX-Weather-2014T (PHOENIX14T) and MedASL, our newly developed medical-domain dataset. A comparative analysis of translation quality on PHOENIX14T shows that ADAT improves BLEU-4 by 0.02 absolute points and ROUGE-L by 0.11 compared to the Encoder-Decoder Transformer. On MedASL, ADAT surpasses the Encoder-Decoder Transformer with 0.01 absolute points in BLEU-4 and 0.02 in ROUGE-L. For runtime efficiency on MedASL, ADAT reduces training time by 50% and lowers both edge-cloud communication and overall end-to-end system times by 2%. These findings demonstrate that the proposed system offers a precise and efficient SLMT with low communication latency, establishing a foundation for responsible deployment in real-world domains such as healthcare, education, and public services.

Keywords: artificial intelligence; assistive technology; blockchain; cloud computing; computer vision; deep learning; edge computing; natural language processing; neural machine translation; sign language translation; transformers

1. Introduction

More than 430 million people are Deaf or Hard-of-Hearing (DHH) worldwide; this figure is expected to exceed 700 million by 2050 [1]. While DHH individuals communicate using sign language, there is a shortage of certified interpreters. For instance, the ratio of DHH individuals to interpreters in the United States is 4,800:1 [2]. The situation is more severe in the least developed countries, where there are fewer interpreters [3]. This disparity underscores the need for automated and efficient sign language machine translation (SLMT) systems.

SLMT systems are Assistive Technologies (AT) that assist the DHH to overcome communication barriers and foster inclusion and independence, particularly when a human interpreter is unavailable [4,5]. These systems must be real-time, reliable, and secure [6]. Developing such systems can save lives in natural and medical emergencies [7]. Consequently, building an accurate and real-time software-based SLMT system is crucial.

 doi:10.20944/preprints202510.0130.v1

Several works investigate SLMT systems [8–18]. However, these systems consider only a few components of the building blocks of an end-to-end privacy-preserving, real-time, and efficient SLMT. In this paper, we fill this gap by proposing an end-to-end SLMT system that considers deployment factors, including the computational costs of the different underlying components, the various system components required for building a real-time system, as well as privacy requirements. In particular, we propose an end-to-end system that integrates five components: (1) A sign language recognition (SLR) module to capture sign videos, (2) An artificial intelligence (AI)-enabled application that serves as a gateway between the user and the edge, (3) Edge nodes to extract and preprocess keypoints from the sign videos for inference, (4) Cloud servers to develop and deploy the SLMT AI model, and (5) Blockchain [19] to record and manage user consent for data collection, ensuring compliance with relevant regulations [20,21]. We address user privacy through consent mechanisms, which we categorize into 1) *system-level* consents in which the DHH allows systems administrators to access user data for system failure diagnosis and recovery, and 2) *application-level* consents to obtain users' permission on data sharing and privacy management. To evaluate the proposed system in a real-world setup, we conduct a comparative analysis of the Encoder-Decoder Transformer [22] and our proposed adaptive Transformer (ADAT) [23], using the RWTH-PHOENIX-Weather-2014T (PHOENIX14T) [24] and MedASL datasets that we created to exemplify conversation with a medical professional.

The main contributions of this paper are as follows:

- We propose an end-to-end edge–cloud–blockchain system for SLMT.
- We evaluate our proposed system in comparison with the most used Encoder-Decoder Transformer using the largest publicly available German sign language dataset, PHOENIX14T, and our new MedASL dataset.
- We conduct experiments and numerical analysis of our system's performance in terms of Bilingual Evaluation Understudy (BLEU), Recall Oriented Understudy for Gisting Evaluation (ROUGE), training time, translation time, and overall end-to-end system time.
- We deploy the Encoder-Decoder Transformer and ADAT models in a unified setup, demonstrating the feasibility of our proposed system for real-world applications.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 explains the proposed end-to-end edge-cloud-blockchain architecture for SLMT. Section 4 describes the materials and methods. Numerical experiments and comparative results are provided in Section 5. Section 6 discusses the responsible deployment of the proposed system. Finally, Section 7 concludes the paper.

2. Related Work

Several works applied machine and deep learning models for SLMT [8–18]. Table 1 summarizes these works by comparing translation formulation, input modality, preprocessing techniques, and learning algorithms. The table also compares the deployment environments, including edge, cloud, and blockchain, as well as consent mechanisms and runtime measurements.

Table 1. Comparison between sign language machine translation systems in the literature.

		Translation Formulation			Model		Deployment Environment			Consent		Runtime Measurement			
Work	Sign Language(s)	Type	Unit	Input Modality	Preprocessing	Algorithm	Edge	Cloud	Blockchain	System	Application	Training time	Preprocessing time	Inference time	Communication time
[8]	ASL	S2G, S2T	Words + Sentences	Keypoints	Smoothing and normalization	Hierarchical Bidirectional RNN	✓	✗	✗	✗	✓	✗	✗	✓	✗

 doi:10.20944/preprints202510.0130.v1

[9]	ASL	S2G2T	Words + Sentences	RGB	Segmentation and contour extraction	Attention-based LSTM	✓	✗	✗	✗	✗	✗	✗	✗	✗
[10]	SSL	S2G	Words	RGB	Segmentation, contour extraction, ROI formation, and normalization	CNN + Tree data structure	✓	✗	✗	✗	✗	✗	✗	✓	✗
[11]	DGS, CSL	S2G, S2G2T	Sentences	RGB	Greyscale, resizing, and ROI formation	GCN + Transformer decoder	✓	✗	✗	✗	✗	✗	✗	✓	✗
[12]	DGS, CSL	S2G, S2G2T	Sentences	RGB	Cropping	Transformer	✗	✗	✗	✗	✗	✗	✗	✗	✗
[13]	ASL, BSL, JSL, KSL	S2G	Words	RGB	Resizing and segmentation	Attention-based CNN	✗	✗	✗	✗	✗	✗	✗	✗	✗
[14]	ASL, ArSL, KSL	S2G	Characters	RGB	Resizing	MLP	✗	✗	✗	✗	✗	✗	✗	✗	✗
[15]	ASL, TSL	S2G	Characters + Words	RGB	Frame extraction, grayscale conversion, normalization, background subtraction, and segmentation	Attention-based CNN + LSTM	✗	✗	✗	✗	✗	✗	✗	✓	✗
[16]	LSA	S2G	Words	RGB	Segmentation, rescaling	LSTM	✗	✗	✗	✗	✗	✗	✗	✗	✗
[17]	DGS, CSL	S2G	Sentences	Keypoints	Partwise partitioning, and graph construction	Graph Fourier Learning	✓	✗	✗	✗	✗	✗	✗	✓	✗
[18]	ASL	S2G	Words	Keypoints	ROI formation, hand cropping, rescaling, and padding	YOLO	✓	✓	✗	✗	✗	✓	✗	✓	✗
This Work	**DGS, ASL**	**S2G2T**	**Sentences**	**Keypoints**	**Normalization, rescaling, and padding**	**Transformer-Based**	✓	✓	✓	✓	✓	✓	✓	✓	✓

ASL: American Sign Language; ArSL: Arabic Sign Language; BSL: Bangla Sign Language; CSL: Chinese Sign Language; DGS: German Sign Language; JSL: Japanese Sign Language; KSL: Korean Sign Language; LSA: Argentine Sign Language; SSL: Sinhala Sign Language; TSL: Turkish Sign Language
CNN: Convolution Neural Network; GCN: Graph Convolution Network; LSTM: Long Short-Term Memory; RNN: Recurrent Neural Network; ROI: Region of Interest; ✓: included; ✗: not included.

It shows that current works focus on sign-to-gloss (S2G) translation [8,10–18], where a gloss is a written representation of a sign. In an end-to-end SLMT system, sign videos are translated either into gloss then text (S2G2T) or directly into text (S2T). Therefore, S2G, without text output, offers limited benefits for real-world applications [6]. In terms of input modalities, previous studies relied mainly on RGB video [9–16]. While these inputs capture rich visual information such as facial expressions and environmental details, they are sensitive to environmental factors, including lighting, background, and signer features and appearance [25]. These variations can degrade the model's performance and require extensive preprocessing and computationally intensive backbones for feature extraction compared to keypoints [26]. In addition, RGB inputs preserve identifiable features and contextual details that raise privacy concerns, such as re-identification and unintended data exposure [27]. In contrast, the use of keypoints in our proposed system retains abstract visual content, including skeletal landmarks and poses. This reduces identifiable details while preserving essential spatiotemporal features for sign language understanding. Therefore, keypoints reduce computational cost and mitigate privacy risks [28].

The trade-offs between computational costs and privacy implications highly impact the deployment choices. This can be addressed by adopting architectures that integrate edge processing with cloud support and governance mechanisms using blockchain. However, few works explore these architectures partially, focusing on the edge as an offloading component, which limits system

 doi:10.20944/preprints202510.0130.v1

architecture scalability [29]. While many works employ edge devices [8–11,17,18], VisioSLR [18] incorporates cloud servers [30], and no work integrates a privacy-preserving blockchain system [31].

The deployment environment influences runtime behavior, making it necessary to measure training, preprocessing, inference, and communication times. Few works report inference time [8,10,11,15,17,18]. Only VisioSLR [18] measures training time, and no works compute preprocessing and edge-cloud communication times. Moreover, despite the importance of user consent in privacy-sensitive applications, no SLMT works implement system-level consents, while DeepASL [8] reports only application-level consent.

Due to the limited adoption of consent mechanisms in SLMT systems, we provide a cross-domain comparison of both consent mechanisms applications in Table 2. Prior works show that system-level consents preserve privacy and anonymity when reporting platform crashes, such as in browsers [32] and operating systems [33]. On the other hand, application-level consents anonymize personal data in contexts including research participation [34], emergency and crime reporting [35–37], and digital health data sharing [38].

Table 2. Comparison of consent applications across domains.

Work	Application Domain	Use Case	Consent		Data Collected
			Type	Purpose	
[32]	Web browsing	Web browser crash reporting	System	Improve software reliability	Crash/bug-related traces, metadata, and user information
[33]	Operating system	Operating system crash reporting	System	Bug diagnosis and security analysis	Memory snapshots, CPU registers, user credentials, browsing history, cryptographic keys
[34]	Digital behavior	Willingness to share data for research	Application	Participate in research	GPS location, photos/videos (house and self), wearable sensor data
[35]	Critical incident management	Anonymous emergency reporting to authorities	Application	Share data with authorities	Geo-coordinates, status alerts, media (optional)
[36]	Blockchain-based reporting	Anonymous crime reporting to authorities	Application	Report criminal incidents	User data, embedded report data
[37]	Blockchain-based reporting	Anonymous crime reporting to authorities	Application	Report criminal incidents	Reported information and cryptographic keys
[38]	Digital health	Sharing personal health data with healthcare and research entities	Application	Control data access and sharing	Personal health, genomic, medications, mental health data
This Work	**Assistive technology**	**Sign language translation**	**System and Application**	**Improve software reliability, bug diagnosis, enable private communication**	**Videos, system performance logs (optional)**

In summary, prior works on SLMT focused on S2G translation using RGB video inputs, despite their computational overhead and privacy vulnerabilities. Keypoint representations, as proven in other domains, offer a more efficient and privacy-preserving alternative. Nevertheless, comprehensive evaluations of deployment factors and runtime measurements in SLMT remain scarce. Similarly, while cross-domain works highlight the benefits of consent mechanisms for privacy-sensitive applications, such mechanisms remain unexplored in the current SLMT literature. In this work, we address this void by proposing an end-to-end system for SLMT, integrating a camera, an AI-enabled application, edge nodes, cloud servers, and blockchain. This aims to enhance the SLMT system's scalability while ensuring privacy-aware communication.

3. Proposed End-to-End System for Sign Language Translation

The overview of our proposed end-to-end edge-cloud-blockchain system for SLMT is presented in Figure 1. We explain the system components in the following subsections.

 doi:10.20944/preprints202510.0130.v1

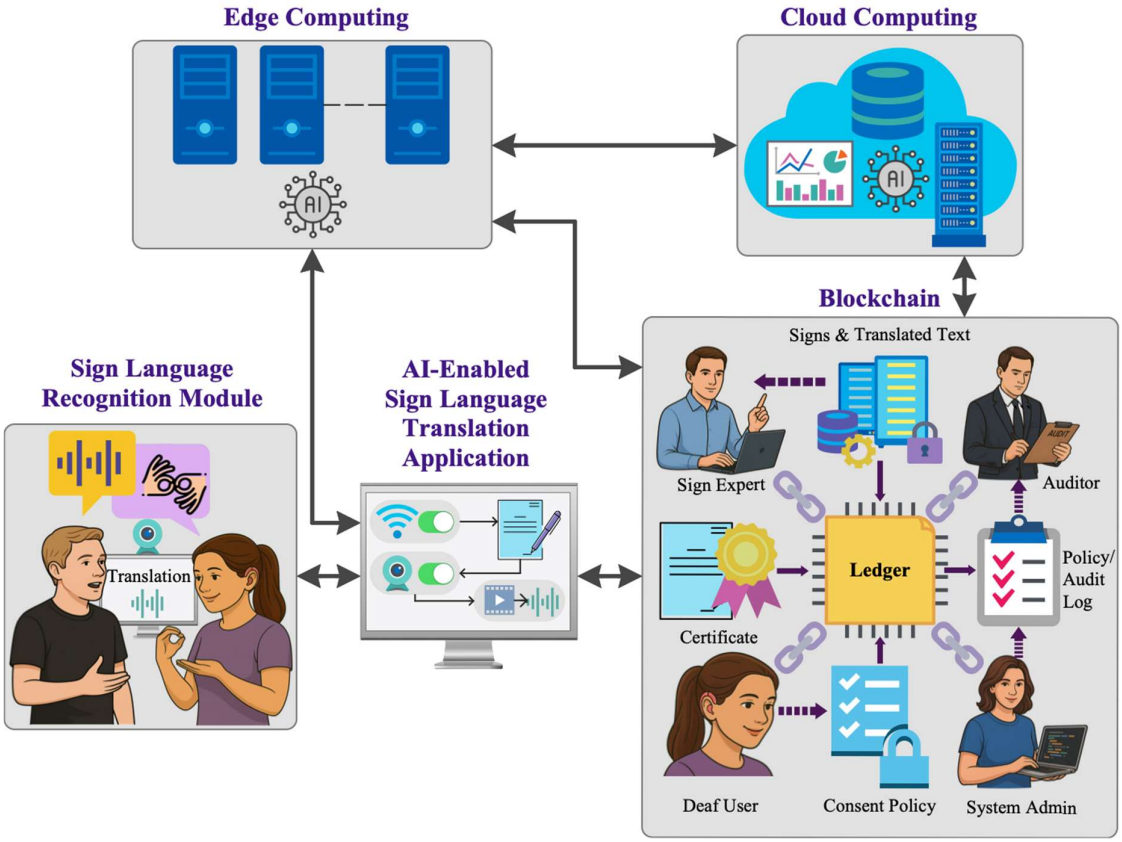

Figure 1. Overview of the proposed end-to-end edge–cloud–blockchain architecture for sign language translation.

3.1. Sign Language Recognition Module

Sign language is based on visual cues, making video capture and feature extraction essential components for translation. The SLR module captures sign videos within a conversation through a camera and transmits them to the edge for keypoint extraction and processing. This enables direct communication between DHH and hearing individuals without reliance on a human interpreter. The technical feasibility and reliable operation of the SLR module depend on the following specifications:

- *Camera configuration:* Higher resolution and frame rate per second improve fine-grained feature extraction [39], but increase computation and video uplink load [26]. Keypoint extraction mitigates this issue by reducing uplink demand and latency.
- *Device computation:* CPU, GPU, and RAM capacities determine the feasibility of capturing frames, extracting keypoints, and performing inference [40].
- *Connectivity:* Wi-Fi and 5G networks enable low-latency and high-throughput communication [41,42].
- *Operating System (OS):* The OS manages camera access, hardware accelerators, security, and user experience. User-driven access controls reduce unauthorized access. Hardware accelerators determine the feasibility of keypoint extraction and inference. Security policies enforce user privacy and implement tamper-evident safeguards, including integrity checks and tamper detection modules [43]. Permissions for camera, network, and storage are tied to user consent, ensuring legal compliance [44].

Our proposed system provides an accurate translation with low latency. CNN-based keypoint extraction achieves state-of-the-art accuracy [45,46]. Therefore, we recommend 1080p at 30 or 60 fps for video streaming, aligning with the public sign language datasets [6,24]. Privacy is ensured

 doi:10.20944/preprints202510.0130.v1

through data minimization, encryption, and revocable consent, all of which are aligned with applicable regulations [20,21].

3.2. AI-Enabled Sign Language Translation Application

The AI-enabled application serves as a user gateway to the system, acquiring sign video input and transmitting it to the edge. The interaction flow proceeds as follows: 1) Enable Wi-Fi or 5G connectivity, 2) establish an authenticated session through credential-based login, 3) obtain user consent, 4) capture sign videos and extract keypoints for preprocessing and inference, and 6) receive and display the translation.

Translation is best performed using Transformer-based models [6,22], which surpass earlier CNN- and RNN-based approaches. This is due to the transformer's ability to capture long-range spatiotemporal dependencies, enabling parallel sequence processing and improving representation learning [22,47].

To ensure transparency, user consents are explained in spoken and sign languages, clearly defining the purpose, data types, and retention periods. Each obtained consent is bound to a policy and an audit log that are immutably stored on the blockchain. Table 3 presents the consents in our end-to-end system.

Table 3. Consent types in the proposed AI-enabled sign language translation application.

Consent Type	Purpose	Required Access	Data Transmitted	Data Storage	Revocation Effect	Status
Application	Translation	Camera	Keypoints and metadata	None	Camera access denied, translation stops	Mandatory
Application	Improvement and retraining	Keypoints and raw sign videos	Raw videos, keypoints, and metadata	On cloud	Immediate, no future data sharing, per policy	Optional
System	Reliability improvement	Raw sign videos	Incident logs, raw videos, and metadata	On cloud	Immediate, no future data sharing, per policy	Optional

3.3. Edge Computing

Once the video streams are transmitted, the edge device performs keypoint extraction and preprocessing. Keypoint extraction is a crucial step as it reduces bandwidth consumption, communication overhead [48], and privacy risks [28]. The edge preprocesses keypoints through normalization, rescaling, and padding, before translating sign language into spoken language using a Transformer-based model. The resulting translation is returned to the application for display to the user.

When an application-level consent for model improvement is granted, the edge forwards the raw videos and the corresponding keypoints and translations to the cloud. Alternatively, under system-level consent, the edge transmits incident logs and associated sign videos and metadata to the cloud. Independent of the consent mechanism, the edge communicates with the blockchain to send records that include consent receipt, system certificate, policy version, and processed and retained data receipts.

3.4. Cloud Computing

The edge communicates with the cloud only when the user grants application-level consent for model improvement and/or system-level consent. Under application-level consent, sign experts annotate uploaded sign videos for future model retraining whenever translation precision falls below predefined thresholds. Updated models are then rolled out to the edge. Under system-level consent, system administrators analyze incident logs and raw videos to diagnose and resolve reliability issues before deploying new updates. In both consent scenarios, the cloud generates immutable records that include the system certificate, retained data receipts, and policy version. These records are

 doi:10.20944/preprints202510.0130.v1

transmitted to the blockchain to ensure auditability without exposing user content. Figure 2 illustrates the deep learning operations pipeline between the edge and cloud in our proposed system.

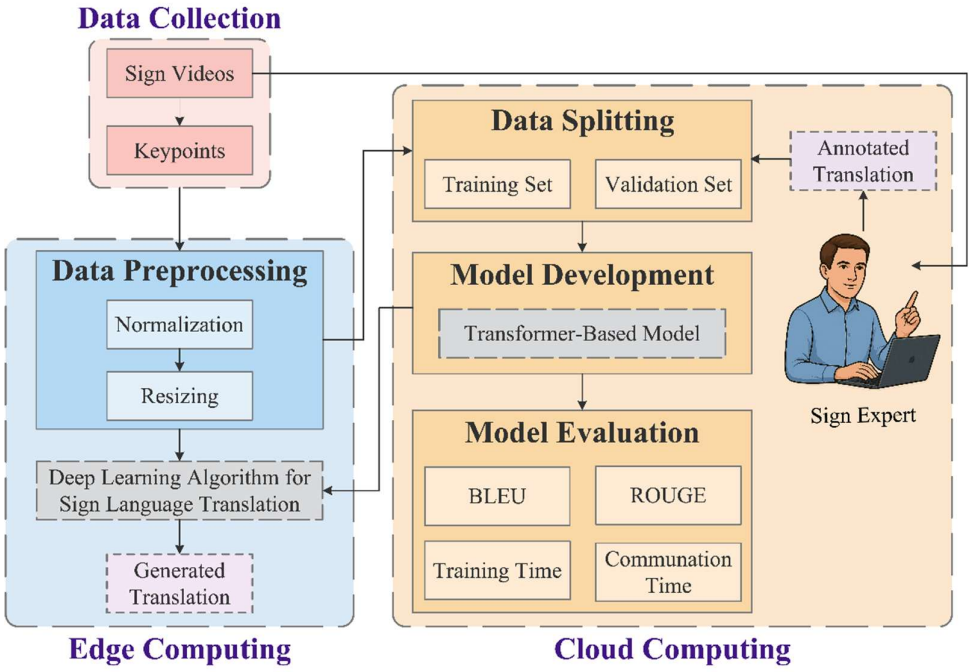

Figure 2. Edge-cloud pipeline for the proposed end-to-end sign language translation system.

3.5. Blockchain

Ensuring the security and privacy of user data is a key requirement in SLMT system deployment. In particular, storing user data on infrastructures, such as the cloud, offers scalability but limited traceability of data access and policy enforcement [49]. To address this issue, our proposed system integrates blockchain as a tamper-evident audit layer for user consents, data access and usage, system certificates, and policies. This retains cloud scalability while improving confidentiality and integrity [50,51].

The blockchain ledger records immutable content-free metadata in the form of consent receipts, data usage receipts, and system certificates. Consent receipts are recorded whenever consent is granted, updated, or revoked. Data usage receipts document how consented data were handled, specifying when it was accessed by experts or administrators, utilized within the translation pipeline, and retained for model improvement. System certificates capture system versions and rollout windows, enabling traceability from training to deployment. Each ledger entry is digitally signed and time-stamped, providing auditors with tamper-evident logs without exposing personal identifiers or sign content. This setup establishes a feedback loop where edge and cloud servers read these entries to enforce policy and maintain integrity. In particular, the edge and cloud retrieve the latest consent receipts and policy commits to authorize or suspend data processing, validate data storage, and verify system certificates before loading or updating the system. Table 4 summarizes the proposed blockchain layer, listing the participants, events, and the associated actions that affect the ledger transactions, log assets, scope, and system effects.

Table 4. Participants, events, transactions, assets, and system effects in the blockchain layer for Sign Language Translation.

Participant	Event	Transaction	Asset	Scope	System Effect
User (Deaf)	Authorize consent	Grant consent	Consent receipt	1. Translation	By scope: 1. Enable translation 2. Allow data storage and access

 doi:10.20944/preprints202510.0130.v1

				2. Model improvement 3. System diagnostics	3. Allow diagnostic uploads and access on the client
	Change consent scope(s)	Update consent	Consent receipt	Any of the above	Adjust allowed data processing, storage, and access on the client
	Withdraw consent	Revoke consent	Consent receipt	Any of the above	Stop processing under the revoked scope on the client
Sign Expert	Request viewing of consented sign videos	Request data access	Data-use receipt: access outcome (granted/ denied)	Curation	Gate and log access on cloud
	Begin sign video annotation	Annotate data (if granted)	Access outcome	Curation	Annotation proceeds on cloud only if access is granted
System Administrator	System/ model passes security/ privacy/ evaluation checks	Deploy system/ model	System certificate	Release system/ model	Roll out new system/ model on edge and cloud
	Policy/ system updated, retrained model improved	Update system/ model version	System certificate	Release new system/ model version	Roll out updated system/ model on edge and cloud
	Policy annulled, vulnerability detected, model performance degraded	Revoke system/ model version	System certificate	Withdraw system/ model version	Roll back to previous system/ model version on edge and cloud
	Approved change in policy	Commit policy	Policy log	Policy	Enforce updated policy on edge and cloud
Auditor	Periodic review	Read audit trails	Audit logs	Governance and legal compliance	Verify compliance and integrity on the ledger

Notes:
- Consent entries include scope and validity window; revocation takes effect immediately, without affecting any previous data collection.
- All transactions are logged on the ledger for auditing.

In summary, the proposed system integrates the SLR module, AI-enabled application, edge, cloud, and blockchain. This design enables scalability, privacy, and reliability while providing accurate and trustworthy sign language translation.

4. Materials and Methods

This section presents the experimental setup and numerical comparative analysis of our proposed system. To evaluate system performance, we employ two datasets covering distinct sign languages and domains, benchmarking the Encoder-Decoder Transformer, the widely adopted algorithm in SLMT, and its variant, ADAT. We assess the performance using BLEU [52] and ROUGE [53] for translation precision, as well as training time, translation time, edge-cloud communication time, and overall system time.

4.1. Datasets

We conduct experiments on PHOENIX14T [24], the most used benchmark dataset for SLMT, and MedASL, an extended version of MedASL [23]. The earlier MedASL release contains 500 annotated video samples; in this work, we expand it to 2,000 samples while following the same scope and methodology. The new recordings were collected in varied environments, including changes in lighting and background, to enhance robustness. MedASL retains the consistent style of medical-related statements as the original dataset. Using a large, multi-signer dataset (PHOENIX14T) and a smaller, single-signer dataset (MedASL) allows comparative analysis of data efficiency, generalization, and end-to-end efficiency across different scales. Table 5 summarizes the characteristics of both datasets.

 doi:10.20944/preprints202510.0130.v1

Table 5. Datasets characteristics.

Dataset	Sign Language	Domain	# Videos	Vocabulary Size		Sentence length		# Signers	Resolution @fps
				Gloss	Text	Gloss	Text		
PHOENIX14T [24]	German	Weather	8,257	1,115	3,004	32	54	9	210 × 260 @25 fps
MedASL	American	Medical	2,000	1,142	1,682	10	16	1	1280 × 800 @30 fps

4.2. Data Preprocessing

We adopt a consistent preprocessing pipeline for training and inference. We use MediaPipe [54] to extract the following per-frame keypoints: 21 landmarks per hand, 468 face landmarks, 5 iris landmarks per eye, and 6 upper body pose landmarks. We then normalize and rescale these keypoints. Next, we concatenate frames belonging to the same sentence into a temporal sequence and zero-pad them for batch processing. During inference, we segment the input stream into sliding windows before passing them to the model.

Regarding the translation output, we construct a vocabulary that includes special tokens, such as start- and end-of-sequence (<SOS> and <EOS>) and <UNK> for unknown words. We then tokenize the text sequences, index each token, and zero-pad them to enable batched training and evaluation.

During evaluation, we use the predefined train/dev/test sets of PHOENIX14T. For MedASL, we allocate 80% for 5-fold cross-validation and 20% for testing.

4.3. Model Development

We evaluate our proposed end-to-end system using two AI models, the mostly used Encoder-Decoder Transformer [22] and ADAT [23].

4.3.1. Encoder-Decoder Transformer

The Encoder-Decoder Transformer [22] is the most used and accurate model for SLMT due to its ability to capture long-range spatiotemporal dependencies [6]. Its encoder–decoder structure relies on self-attention to capture sequences and cross-attention to align the encoder outputs with decoder representations. The attention mechanisms are presented in equations (1)-(3), respectively.

$$Self - Attention = Sofmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

where Q, K, and V represent query, key, and value, respectively. d_k is the dimension.

$$Multi - Head\ Attention = Concat(Attention_1 \dots Attention_i) * w^o \quad (2)$$

where w^o is a learned weight that adds parameters in a single-head self-attention.

$$Cross - Attention(Q_d, K_e, V_e) = Sofmax(\frac{Q_d K_e^T}{\sqrt{d_k}})V_e \quad (3)$$

where Q_d is the decoder query and K_e, V_e are from the encoder output.

4.3.2. Adaptive Transformer (ADAT)

ADAT [23] modifies the Encoder-Decoder Transformer [22] by integrating convolution layers for localized feature extraction, LogSparse Self-Attention (LSSA) [55] to reduce the quadratic cost by computing a logarithmically spaced subsets of past frames, and an adaptive gating mechanism [56] that balances short- and long-range dependencies by weighting the outputs of LSSA and Global Average Pooling (GAP). Equations (4) and (5) show LSSA and the gating mechanism, respectively.

$$LSSA(Q, K) = Sofmax\left(\frac{Q_{I_p^j} K_{I_p^j}^T}{\sqrt{\frac{d_k}{2}}}\right) \quad (4)$$

where I_p^j is the patch indices that a current patch p can attend during a logarithmic computation from j to $J + 1$.

$$\boldsymbol{Gating = g \cdot LSSA + (1 - g) \cdot GAP} \quad (5)$$

 doi:10.20944/preprints202510.0130.v1

where g is the gate value computed via a gate neural network.

Prior studies have shown that the adaptive gating mechanism achieves comparable or superior translation quality to the Encoder-Decoder Transformer while significantly improving efficiency and scalability [23,57].

Evaluating the Encoder-Decoder Transformer and ADAT underscores the trade-off between translation quality and computational cost, providing insights into the suitability of each architecture for real-world SLMT deployment.

4.4. Experimental Setup

The proposed system operates in two stages: 1) S2T translation, executed by the AI-enabled application in coordination with the edge device, and 2) SLMT development, which trains AI models on benchmark datasets and retrains them on consented, annotated user videos.

In the first stage, sign videos are captured at 30 frames per second using an Intel RealSense D455 camera, connected to the edge, which is a desktop in our experiments. They are streamed to the application. The edge device extracts keypoints via MediaPipe, applies preprocessing, and performs inference. In the second stage, the edge connects to a cloud server for model training. Retrained models are periodically deployed back to the edge for updated inference. Figure 3 illustrates the experimental setup, where a user signs in ASL in front of an Intel RealSense D455 camera connected to the edge device.

(**a**) (**b**)

Figure 3. Experimental setup with a user signing in ASL. The video feed with overlaid keypoints is displayed on screen. (**a**) A status message is displayed during inference; (**b**) The translated text is displayed once inference is complete.

4.5. Experiments

We evaluate the translation quality and runtime efficiency of the proposed system under scenarios that mirror real-world system operation. During training and validation, we measure the quality of generated translations against human-annotated references using BLEU [52] and ROUGE [53] metrics. These metrics provide insights on translation precision and coverage. Applying them across both datasets enables direct comparison between the Encoder–Decoder Transformer and ADAT architectures.

Regarding efficiency, during training, we log the time required for model convergence. During inference, we record preprocessing and model inference times on the edge device. In addition, we measure the overall end-to-end system time experienced by the user, which combines preprocessing, inference, and edge–cloud communication. During edge-cloud interaction, we capture the uplink and downlink times associated with transferring user data and model deployment. These experiments provide the methodological basis for the runtime evaluation in Section 5.

5. Performance Evaluation

 doi:10.20944/preprints202510.0130.v1

This section evaluates the models under study within a unified experimental setup, focusing on translation quality and runtime efficiency. We structure the evaluation into four components: the metrics used to assess translation and runtime, the hyperparameter tuning strategy, the hardware specifications for training and inference, and an in-depth numerical analysis of the obtained results.

5.1. Evaluation Metrics

To ensure a comparable assessment of linguistic accuracy and system-level efficiency, we measure translation quality using BLEU and ROUGE and assess runtime efficiency by reporting training, translation, communication, and end-to-end system times.

5.1.1. Bilingual Evaluation Understudy (BLEU)

We use a normalized BLEU score with 1 to 4 n-grams, as shown in equations (6)-(7).

$$BLEU = BP \cdot e^{(\sum_{n=1}^{N} w_n \log(p_n)} \quad (6)$$

where p_n is the precision of n-grams, w_n is the weight of each n-gram size, and BP is the Brevity Penalty.

$$BP = \begin{cases} 1, if\ G > r \\ e^{(1-\frac{r}{G})}, if\ G \leq r \end{cases} \quad (7)$$

where G is the length of the generated translation. r is the reference corpus length.

5.1.2. Recall Oriented Understudy for Gisting Evaluation (ROUGE)

We assess the similarity between generated translations and their references using ROUGE-1 and ROUGE-L. ROUGE-1 measures unigram overlap, while ROUGE-L assesses the longest sequence of overlapping words. We normalize both metrics through precision and recall. ROUGE-L is calculated using equations (8)-(10).

$$ROUGE_{precision} = \frac{\sum_{n=1}^{N} LCS(G,R)}{\sum_{n=1}^{N} |G|} \quad (8)$$

$$ROUGE_{recall} = \frac{\sum_{n=1}^{N} LCS(G,R)}{\sum_{n=1}^{N} |R|} \quad (9)$$

where G is the generated translation. R is the reference translation. $LCS(G, R)$ is the length of the longest common subsequence between G and R. $|\cdot|$ is the number of tokens in the translation.

$$ROUGE - L = \frac{2 \cdot ROUGE_{precision} \cdot ROUGE_{recall}}{ROUGE_{precision} + ROUG_{recall}} \quad (10)$$

5.1.3. Runtime Efficiency

We measure the runtime efficiency by calculating the training time and the latency across training, preprocessing, inference, translation, edge-cloud communication, and the overall end-to-end system time, as presented in Table 6. These measurements capture the user-perceived latency and the backend operational overhead.

Table 6. Runtime efficiency metrics, definitions, and equations.

Metric	Notation	Definition	Equation
Training time	T_{train}	Time required for model convergence during training.	$T_{\text{train}} = N_{epochs} \times T_{epoch}$
Preprocessing time	T_{pre}	Time required to convert sign videos into keypoints and prepare them for inference.	Not applicable
Inference time	T_{infer}	Time required for model inference once preprocessing is completed.	Not applicable
Translation time	$T_{\text{translate}}$	Total user-perceived translation time.	$T_{\text{translate}} = T_{pre} + T_{\text{infer}}$
Edge-cloud communication time[1]	$T_{\text{edge2cloud}}$	Uplink latency for transmitting keypoints and videos from the edge to the cloud.	Not applicable
Cloud-edge communication time[2]	$T_{cloud2edge}$	Downlink latency for transmitting trained models from the cloud to the edge.	Not applicable
Overall end-to-end system time	$T_{end2end}$	Total system latency.	$T_{\text{end2end}} = T_{\text{translate}} + T_{\text{edge2cloud}}$

 doi:10.20944/preprints202510.0130.v1

[1] Occurs under application-level consent for model improvement and/or system-level consent.
[2] Occurs periodically after model retraining.

5.2. Hyperparameter Tuning

We perform systematic hyperparameter tuning to produce the best translation in both models. The search space includes optimization strategies, training setups, model architectures, and decoding choices. The final hyperparameters are selected based on validation BLEU-4 scores with early stopping. Table 7 summarizes the hyperparameters investigated, the ranges explored, and the chosen settings.

Table 7. Hyperparameter search space and selected configurations.

Category	Hyperparameter	Configurations Studied	Selected Configurations
Optimization	Loss functions	Gloss: CTC loss, none Text: Cross-Entropy loss	Gloss: CTC loss Text: Cross-Entropy loss
	Label smoothing	0, 0.01, 0.05, 0.1	0.1
	Optimizer	Adam with β_1=0.9 (default) or 0.99 and β_2=0.999 (default) or 0.98	Adam with β_1=0.9, β_2=0.98
	Learning rate	5×10^{-5}, 1×10^{-5}, 1×10^{-4}, 1×10^{-3}, 1×10^{-2}	5×10^{-5}
	Weight decay	0, 0.1, 0.001	0
	Early stopping	Patience = 3, 5 Tolerance = 0, 1×10^{-2}, 1×10^{-4}, 1×10^{-6}	Patience = 5 Tolerance = 1×10^{-6}
	Learning rate scheduler	Constant with linear warmup until either 1,000 steps have been taken or 5% of the total training steps are reached, whichever is larger.	
Training setup	Gradient clipping	None, 0.1, 0.5, 1.0, 2.0, 3.0, 4.0, 5.0	5.0
	Epochs	30, 100	30
	Batch size	32	32
Model architecture	Embedding dimension	256, 512	512
	Encoder layers	1, 2, 4, 12	1
	Decoder layers	1, 2, 4, 12	1
	Attention heads	8, 16	8
	Feed-forward size	512, 1024	512
	Dropout	0, 0.1	0.1
Decoding	Gloss	Greedy or beam search	Greedy
	Text	Greedy or beam search	Beam search with a size of 5 with reranking using a 4-gram KenLM * (interpolation weight 0.4, length penalty 0.2).

CTC: Connectionist Temporal Classification [58].
*: We build a KenLM model [59] using text tokenized with SentencePiece [60] into BPE [61] sub-word units to rerank the generated translation.

5.3. Hardware Specifications

The evaluation was conducted using a distributed setup consisting of capture, edge, and cloud devices. The capture unit provides lightweight video acquisition, the edge device handles preprocessing and inference, and the cloud server supports model development and data storage. This division reflects practical deployment, where time-sensitive tasks are executed closer to the user while high-compute workloads are offloaded to the cloud. Table 8 lists the specifications of each device. All software components are implemented using Python 3.10.

Table 8. Hardware specifications for performance evaluation.

Device	CPU	GPU	Storage	RAM	Connectivity	Operating System
Intel RealSense Camera	Vision Processor D4 (ASIC)	NA	2 MB	NA	USB Type-C	NA
Edge	1 x Intel Core i7-8700 at 4.20 GHz, 6 cores/CPU	NA	475 GB	8 GB	Ethernet	Windows

 doi:10.20944/preprints202510.0130.v1

Cloud	2 x AMD EPYC 7763 at 2.45 GHz, 64 cores/CPU	2 x NVIDIA RTX A6000 48GB/GPU	5 TB	1 TB	Wi-Fi, LTE*, Ethernet	Ubuntu

NA: Not available; *: Supported through external hotspot.

5.4. Experimental Results Analysis

This section presents a numerical evaluation of the Encoder-Decoder Transformer and ADAT in terms of translation quality and runtime efficiency.

5.4.1. Translation Quality

Table 9 compares the translation quality of the Encoder-Decoder Transformer and ADAT models across PHOENIX14T and MedASL datasets. On PHOENIX14T, ADAT surpasses the Encoder-Decoder Transformer in all metrics, with +0.02 and +0.11 absolute points in BLEU-4 and ROUGE-L, respectively. Similarly, on MedASL, ADAT outperforms the Encoder-Decoder Transformer with +0.01 and +0.02 in BLEU-4 and ROUGE-L, respectively. These results underscore ADAT's ability to preserve longer semantic overlap with reference translations through its attention design and adaptive feature weighting [23].

Table 9. Comparison of models for sign language translations (scores range between 0 and 1; higher being better).

Model	Dataset	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-1	ROUGE-L
Encoder-Decoder Transformer	PHOENIX14T	0.32	0.21	0.14	0.10	0.18	0.16
ADAT		**0.33**	**0.23**	**0.16**	**0.12**	**0.28**	**0.27**
Encoder-Decoder Transformer	MedASL	0.39	0.24	0.15	0.09	0.19	0.18
ADAT		**0.43**	**0.26**	**0.16**	**0.10**	**0.20**	**0.20**

The unbalanced gains on ROUGE-L compared to BLEU-4 highlight ADAT's design in capturing global spatiotemporal structure rather than enhancing local sentence precision. BLEU-4 emphasizes exact matches of 4-gram sequences; therefore, the observed results reflect marginal improvements in local precision. In contrast, ROUGE-L captures the overall sentence structure and semantic consistency. Consequently, our results show larger improvements at the sentence level.

Moreover, the difference between the datasets further illustrates this trend. PHOENIX14T has longer and more structured sentences, where maintaining global information order is essential. Conversely, MedASL sentences are shorter and more diverse, which limits the improvement in overall sentence-level structure. Accordingly, ADAT achieves a +0.11 gain in ROUGE-L on PHOENIX14T but only a +0.02 gain on MedASL.

In summary, our experimental results demonstrate that while ADAT's BLEU score gains are minimal, ROUGE-L score confirms its effectiveness in SLMT. In particular, ADAT is designed to preserve sentence-level organization and global semantic coherence. Therefore, it delivers translations that maintain the overall structure and meaning of the original message, making it a more effective model for real-world SLMT systems.

5.4.2. Runtime Efficiency

We assess the models' runtime efficiency within the edge–cloud pipeline using the MedASL dataset. Figure 4 presents the model training time in both datasets. Table 10 presents the runtime and system performance for both models. Figure 5 illustrates ADAT's relative improvements over the Encoder-Decoder Transformer.

 doi:10.20944/preprints202510.0130.v1

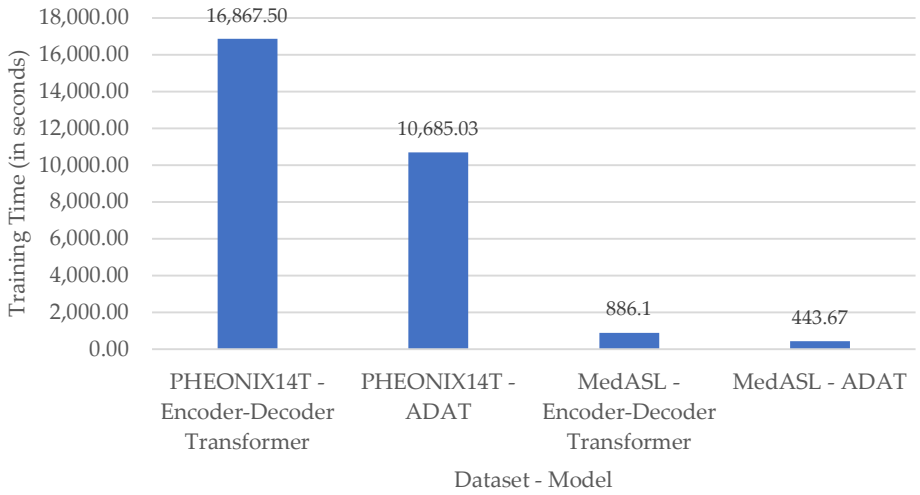

Figure 4. Comparison of sign language training time for the models under study in our proposed translation system.

Table 10. Comparison of sign language translation runtime and system performance.

Model	Preprocessing Time (ms)	Inference Time (ms)	Translation Time (ms)	Cloud-to-Edge Communication Time (ms)	Edge-to-Cloud Communication Time (ms)	Overall End-to-End System Time (ms)	Model Size (MB)	Keypoints Size (MB)
Encoder-Decoder Transformer	0.03	188.25	188.28	11,725.42	324.09	512.37	134.8	3
ADAT	**0.02**	**183.75**	**183.77**	**11,517.81**	**312.85**	**496.62**	**104.9**	3

ms: milliseconds.

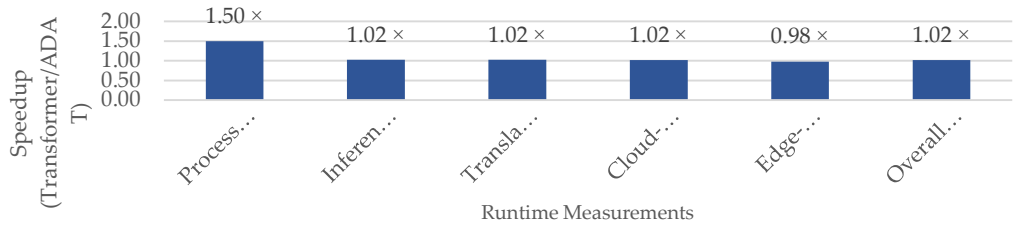

Figure 5. Relative improvement by component (>1 being ADAT is faster).

The observed results indicate that training on PHOENIX14T requires a substantially longer time than MedASL due to its larger size and vocabulary. Within each dataset, ADAT consistently converges faster than the Encoder–Decoder Transformer, reflecting its reduced computational overhead. In particular, ADAT reduces training time by 1.6× on PHOENIX14T and by half on MedASL, due to its adaptive attention mechanism [23].

In real-world deployment, translation and communication times are critical constraints. In our experiments, ADAT slightly decreases translation time from 188.28 ms to 183.77 ms through faster inference, resulting in a 1.02× relative improvement. It also achieves 1.02x speedup in cloud-edge communication. This is explained by ADAT's 22% smaller model size compared to the Encoder-Decoder Transformer (104.9 MB vs. 134.8 MB), which lowers computational overhead [23]. When considering the complete pipeline, ADAT delivers a 1.02× reduction in overall end-to-end time,

 doi:10.20944/preprints202510.0130.v1

showing that latency gains remain consistent across the entire pipeline. These findings confirm that ADAT sustains efficiency advantages during training and in distributed environments.

In summary, ADAT combines a significant reduction in training cost with speedup in translation and data transfer. These characteristics make ADAT well-suited for deployment in SLMT systems operating in distributed edge–cloud environments with blockchain integration.

6. Responsible Deployment of the Proposed System

The experimental results, presented and discussed in Section 5, demonstrate the technical feasibility of SLMT systems in an edge–cloud architecture. However, real-world SLMT deployment requires specific considerations, which we address in this section.

6.1. Ethical, Legal, and Data Rights Considerations

Sign language video recordings contain sensitive biometric and contextual information. As outlined in the Experimental Setup (Section 4.4), the deployed system captures high-resolution videos and extracts and stores the signer's keypoints for inference and model retraining. Managing such data requires clear protocols for ownership, consent, and accountability, guided by international and national frameworks, such as the General Data Protection Regulation (GDPR) [20] and the UAE Federal Decree Law concerning the Protection of Personal Data (PDPL) [21].

Users must grant consent that they completely understand and can withdraw it at any time. These consents must be auditable to ensure accountability and transparency. The blockchain ledger, described in Section 3.5, provides an immutable record for each consent-related event [50]. Once logged, entries cannot be altered or erased; instead, any changes are appended as new records. This design preserves the whole history of modifications and prevents record tampering.

Furthermore, users should retain ownership of raw video recordings and extracted keypoints, while also having the right to withdraw. Moreover, the collected data must not be repurposed for unrelated applications without renewed consent. In addition, transparency is ensured through blockchain-based logs, providing verifiable and tamper-resistant records of data usage [19].

6.2. Security Threats and Mitigations

An edge–cloud–blockchain SLMT pipeline introduces security risks that must be addressed to ensure secure and reliable deployment. Video tampering poses a significant risk, as it could lead to consequences including consent violations, loss of user trust, and model corruption during retraining [62,63]. This can be mitigated by cryptographic hashing of uploaded samples combined with blockchain-backed integrity checks [64]. In addition, the risk of data manipulation, such as in consent logs, is addressed by the blockchain immutability [50]. Moreover, access to logs and trained models must be protected through strong encryption and multi-factor authentication at both the edge and cloud levels [65]. Integrating these safeguards preserves the efficiency gains reported in Section 5 and maintains security and reliability.

6.3. Minimum Requirements for Sign Language Translation System Deployment

The empirical evaluation and the ethical and security considerations discussed above define the following minimum requirements for deploying SLMT systems:

- *Translation quality* must be sufficient to support transparent and precise communication between DHH individuals and the broader society. Section 5.1 shows that ADAT consistently outperforms the Encoder-Decoder Transformer across all metrics, demonstrating the feasibility of achieving precise translation.
- *Latency* must be minimized to ensure an efficient end-to-end translation. Section 5.2 demonstrates ADAT's reductions in inference and communication times in the edge–cloud setup.

 doi:10.20944/preprints202510.0130.v1

- *Generalization* across signers and environmental conditions is essential to ensure inclusivity. Achieving this requires continuous data collection from different signers, domains, and environments to improve robustness.
- *Data collection and model retraining* must comply with international and national regulations, such as GDPR [20] and UAE PDPL [21]. These principles ensure that users have control over their data, can withdraw their consent, and determine how their data is used. The blockchain-based logging mechanism described in Section 3.5 provides a solid foundation for meeting these obligations and ensuring accountability.
- *Security and robustness* are critical for reliable deployment. As discussed in Section 6.2, safeguards against video tampering and unauthorized access must be integrated into the system design.

In summary, the integration of a precise AI model within an edge–cloud–blockchain architecture demonstrates that advances in translation and runtime efficiency must be accompanied by ethical, legal, and security safeguards. This alignment is essential for deployment in domains such as healthcare, education, and public services, where translation quality and data privacy are critical. By adhering to established regulations, incorporating blockchain-based auditability, and embedding robust security protections, the proposed architecture serves as a blueprint for future SLMT technologies that strike a balance between technical effectiveness and responsible deployment. Beyond SLMT, this work also provides a transferable model for the responsible deployment of AI in other biometric domains, including speech and facial recognition.

7. Conclusions

In this work, we present a novel end-to-end sign language machine translation (SLMT) system built on an edge-cloud-blockchain architecture. The proposed system consists of a sign language recognition (SLR) module, an AI-enabled SLMT application, edge nodes, cloud servers, and a blockchain layer. The SLR module uses a camera to capture sign videos which are then transmitted to the edge. At the edge, keypoints are extracted and preprocessed, and inference is executed locally using a model retrieved from the cloud. Based on user consent, the edge may also upload the keypoints to the cloud for storage and future model development. The blockchain layer provides immutable logging, ensuring compliance with data protection regulations, such as the General Data Protection Regulation (GDPR) and the UAE Federal Decree Law concerning the Protection of Personal Data (PDPL).

Within this framework, we develop and evaluate the Encoder-Decoder Transformer and Adaptive Transformer (ADAT) in terms of translation quality and runtime efficiency. We conduct a comparative analysis on RWTH-PHOENIX-Weather-2014T (PHOENIX14T) and MedASL, an extended medical-domain dataset introduced in this work. The results show that ADAT consistently outperforms the Encoder-Decoder Transformer in translation accuracy and achieves a twofold reduction in training time, with additional gains in inference and communication efficiency. Furthermore, we identify and discuss the minimum requirements for a responsible real-world deployment of SLMT systems to balance computation efficiency, translation quality, and user privacy.

Future work may progress in several directions. Building large, multilingual sign language datasets that incorporate diverse signers across various domains is essential to improve generalization. Integrating multimodal inputs with advanced feature extraction offers the potential for enhancing semantic alignment and translation. Exploring lightweight models can achieve lower latency on constrained edge devices. Finally, extending blockchain towards federated consent and cross-institutional governance may enable a broader adoption of SLMT systems in various domains, such as healthcare, where transparency, accountability, and equity are crucial.

Author Contributions: Conceptualization, L.I.; methodology, L.I..; software, N.S. and L.I.; validation, N.S. and L.I. formal analysis, N.S. and L.I.; investigation, N.S.; resources, L.I.; data curation, L.I.; writing—original draft preparation, N.S.; writing—review and editing, L.I.; visualization, L.I. and N.S.; supervision, L.I.; project

 doi:10.20944/preprints202510.0130.v1

administration, L.I.; funding acquisition, L.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Emirates Center for Mobility Research, United Arab Emirates University, grant number 12R126.

Data Availability Statement: Our proposed MedASL dataset and the implementation code will be made publicly available at https://github.com/INDUCE-Lab/.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. World Health Organization Ageing and Health Available online: https://www.who.int/news-room/fact-sheets/detail/ageing-and-health (accessed on 21 August 2025).
2. Registry of Interpreters for the Deaf Registry Available online: https://rid.org (accessed on 22 August 2025).
3. Kaula, L.T.; Sati, W.O. Shaping a Resilient Future: Perspectives of Sign Language Interpreters and Deaf Community in Africa. *Journal of Interpretation* 2025, *13*.
4. Ismail, L.; Zhang, L. *Information Innovation Technology in Smart Cities*; 2018;
5. Shahin, N.; Watfa, M. Deaf and Hard of Hearing in the United Arab Emirates Interacting with Alexa, an Intelligent Personal Assistant. *Technol Disabil* 2020, *32*, 255–269, doi:10.3233/TAD-200286.
6. Shahin, N.; Ismail, L. From Rule-Based Models to Deep Learning Transformers Architectures for Natural Language Processing and Sign Language Translation Systems: Survey, Taxonomy and Performance Evaluation. *Artif Intell Rev* 2024, *57*, 271, doi:10.1007/s10462-024-10895-z.
7. Tonkin, E. The Importance of Medical Interpreters. *American Journal of Psychiatry Residents' Journal* 2017, *12*, 13–13, doi:10.1176/appi.ajp-rj.2017.120806.
8. Fang, B.; Co, J.; Zhang, M. DeepASL: Enabling Ubiquitous and Non-Intrusive Word and Sentence-Level Sign Language Translation. In Proceedings of the Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems; ACM: New York, NY, USA, November 6 2017; pp. 1–13.
9. S Kumar, S.; Wangyal, T.; Saboo, V.; Srinath, R. Time Series Neural Networks for Real Time Sign Language Translation. In Proceedings of the 17th IEEE International Conference on Machine Learning and Applications (ICMLA); IEEE, December 2018; pp. 243–248.
10. Dhanawansa, I.D.V.J.; Rajakaruna, R.M.T.P. Sinhala Sign Language Interpreter Optimized for Real – Time Implementation on a Mobile Device. In Proceedings of the 2021 10th International Conference on Information and Automation for Sustainability (ICIAfS); IEEE, August 11 2021; pp. 422–427.
11. Gan, S.; Yin, Y.; Jiang, Z.; Xie, L.; Lu, S. Towards Real-Time Sign Language Recognition and Translation on Edge Devices. In Proceedings of the Proceedings of the 31st ACM International Conference on Multimedia; ACM: New York, NY, USA, October 26 2023; pp. 4502–4512.
12. Zhang, B.; Müller, M.; Sennrich, R. SLTUNET: A Simple Unified Model for Sign Language Translation. In Proceedings of the International Conference on Learning Representations; 2023.
13. Miah, A.S.M.; Hasan, Md.A.M.; Tomioka, Y.; Shin, J. Hand Gesture Recognition for Multi-Culture Sign Language Using Graph and General Deep Learning Network. *IEEE Open Journal of the Computer Society* 2024, *5*, 144–155, doi:10.1109/OJCS.2024.3370971.
14. Shin, J.; Miah, A.S.M.; Akiba, Y.; Hirooka, K.; Hassan, N.; Hwang, Y.S. Korean Sign Language Alphabet Recognition Through the Integration of Handcrafted and Deep Learning-Based Two-Stream Feature Extraction Approach. *IEEE Access* 2024, *12*, 68303–68318, doi:10.1109/ACCESS.2024.3399839.
15. Baihan, A.; Alutaibi, A.I.; Alshehri, M.; Sharma, S.K. Sign Language Recognition Using Modified Deep Learning Network and Hybrid Optimization: A Hybrid Optimizer (HO) Based Optimized CNNSa-LSTM Approach. *Sci Rep* 2024, *14*, 26111, doi:10.1038/s41598-024-76174-7.
16. Huang, J.; Chouvatut, V. Video-Based Sign Language Recognition via ResNet and LSTM Network. *J Imaging* 2024, *10*, 149, doi:10.3390/jimaging10060149.
17. Wei, D.; Hu, H.; Ma, G.-F. Part-Wise Graph Fourier Learning for Skeleton-Based Continuous Sign Language Recognition. *J Imaging* 2025, *11*, 286, doi:10.3390/jimaging11080286.

 doi:10.20944/preprints202510.0130.v1

18. Ismail, L.; Shahin, N.; Tesfaye, H.; Hennebelle, A. VisioSLR: A Vision Data-Driven Framework for Sign Language Video Recognition and Performance Evaluation on Fine-Tuned YOLO Models. *Procedia Comput Sci* 2025, *257*, 85–92, doi:10.1016/j.procs.2025.03.014.
19. Ismail, L.; Materwala, H.; Hennebelle, A. A Scoping Review of Integrated Blockchain-Cloud (BcC) Architecture for Healthcare: Applications, Challenges and Solutions. *Sensors* 2021, *21*, 3753.
20. European Parliament and Council of the European Union *General Data Protection Regulation GDPR*; 2018;
21. Government of United Arab Emirates *Federal Decree by Law No. (45) of 2021 Concerning the Protection of Personal Data*; 2021;
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems; 2017.
23. Shahin, N.; Ismail, L. ADAT: Time-Series-Aware Adaptive Transformer Architecture for Sign Language Translation. *ArXiv* 2025.
24. Camgoz, N.C.; Hadfield, S.; Koller, O.; Ney, H.; Bowden, R. Neural Sign Language Translation. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2018; pp. 7784–7793.
25. Kwak, J.; Sung, Y. Automatic 3D Landmark Extraction System Based on an Encoder–Decoder Using Fusion of Vision and LiDAR. *Remote Sens (Basel)* 2020, *12*, 1142, doi:10.3390/rs12071142.
26. Wu, Y.-H.; Liu, Y.; Xu, J.; Bian, J.-W.; Gu, Y.-C.; Cheng, M.-M. MobileSal: Extremely Efficient RGB-D Salient Object Detection. *IEEE Trans Pattern Anal Mach Intell* 2022, *44*, 10261–10269, doi:10.1109/TPAMI.2021.3134684.
27. Climent-Pérez, P.; Florez-Revuelta, F. Protection of Visual Privacy in Videos Acquired with RGB Cameras for Active and Assisted Living Applications. *Multimed Tools Appl* 2021, *80*, 23649–23664, doi:10.1007/s11042-020-10249-1.
28. Ahmad, S.; Morerio, P.; Del Bue, A. Event Anonymization: Privacy-Preserving Person Re-Identification and Pose Estimation in Event-Based Vision. *IEEE Access* 2024, *12*, 66964–66980, doi:10.1109/ACCESS.2024.3399539.
29. Materwala, H.; Ismail, L.; Shubair, R.M.; Buyya, R. Energy-SLA-Aware Genetic Algorithm for Edge–Cloud Integrated Computation Offloading in Vehicular Networks. *Future Generation Computer Systems* 2022, *135*, 205–222, doi:https://doi.org/10.1016/j.future.2022.04.009.
30. L. Ismail; B. Mills; A. Hennebelle A Formal Model of Dynamic Resource Allocation in Grid Computing Environment. In Proceedings of the Proceedings of the 2008 9th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing; 2008.
31. Ismail, L.; Materwala, H.; Khan, M.A. Performance Evaluation of a Patient-Centric Blockchain-Based Healthcare Records Management Framework. In Proceedings of the Proceedings of the 2020 2nd International Electronics Communication Conference; 2020; pp. 39–50.
32. Satvat, K.; Shirvanian, M.; Hosseini, M.; Saxena, N. CREPE: A Privacy-Enhanced Crash Reporting System. In Proceedings of the Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy; ACM: New York, NY, USA, March 16 2020; pp. 295–306.
33. Kim, S.-J.; You, M.; Shin, S. CR-ATTACKER: Exploiting Crash-Reporting Systems Using Timing Gap and Unrestricted File-Based Workflow. *IEEE Access* 2025, *13*, 54439–54449, doi:10.1109/ACCESS.2025.3553746.
34. Struminskaya, B.; Toepoel, V.; Lugtig, P.; Haan, M.; Luiten, A.; Schouten, B. Understanding Willingness to Share Smartphone-Sensor Data. *Public Opin Q* 2021, *84*, 725–759, doi:10.1093/poq/nfaa044.
35. Qureshi, A.; Garcia-Font, V.; Rifà-Pous, H.; Megías, D. Collaborative and Efficient Privacy-Preserving Critical Incident Management System. *Expert Syst Appl* 2021, *163*, 113727, doi:10.1016/j.eswa.2020.113727.
36. Zhu, L.; Zhang, J.; Zhang, C.; Gao, F.; Chen, Z.; Li, Z. Achieving Anonymous and Covert Reporting on Public Blockchain Networks. *Mathematics* 2023, *11*, 1621, doi:10.3390/math11071621.
37. Shi, R.; Yang, Y.; Feng, H.; Yuan, F.; Xie, H.; Zhang, J. PriRPT: Practical Blockchain-Based Privacy-Preserving Reporting System with Rewards. *Journal of Systems Architecture* 2023, *143*, 102985, doi:10.1016/j.sysarc.2023.102985.

 doi:10.20944/preprints202510.0130.v1

38. Lee, A.R.; Koo, D.; Kim, I.K.; Lee, E.; Yoo, S.; Lee, H.-Y. Opportunities and Challenges of a Dynamic Consent-Based Application: Personalized Options for Personal Health Data Sharing and Utilization. *BMC Med Ethics* 2024, *25*, 92, doi:10.1186/s12910-024-01091-3.
39. Feichtenhofer, C.; Fan, H.; Malik, J.; He, K. SlowFast Networks for Video Recognition. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision ; 2019; pp. 6202–6211.
40. Lee, J.; Shi, W.; Gil, J. Accelerated Bulk Memory Operations on Heterogeneous Multi-Core Systems. *J Supercomput* 2018, *74*, 6898–6922, doi:10.1007/s11227-018-2589-x.
41. Liu, R.; Choi, N. A First Look at Wi-Fi 6 in Action: Throughput, Latency, Energy Efficiency, and Security. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2023, *7*, 1–25, doi:10.1145/3579451.
42. Agiwal, M.; Abhishek, R.; Saxena, N. Next Generation 5G Wireless Networks: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials* 2016, *18*, 1617–1655, doi:10.1109/COMST.2016.2532458.
43. Mayrhofer, R.; Stoep, J. Vander; Brubaker, C.; Kralevich, N. The Android Platform Security Model. *ACM Transactions on Privacy and Security* 2021, *24*, 1–35, doi:10.1145/3448609.
44. Roesner, F.; Kohno, T.; Moshchuk, A.; Parno, B.; Wang, H.J.; Cowan, C. User-Driven Access Control: Rethinking Permission Granting in Modern Operating Systems. In Proceedings of the 2012 IEEE Symposium on Security and Privacy; IEEE, May 2012; pp. 224–238.
45. Li, Y.; Zhang, S.; Wang, Z.; Yang, S.; Yang, W.; Xia, S.-T.; Zhou, E. TokenPose: Learning Keypoint Tokens for Human Pose Estimation. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision ; 2021; pp. 1313–11322.
46. Gong, C.; Zhang, Y.; Wei, Y.; Du, X.; Su, L.; Weng, Z. Multicow Pose Estimation Based on Keypoint Extraction. *PLoS One* 2022, *17*, e0269259, doi:10.1371/journal.pone.0269259.
47. Camgoz, N.; Koller, O.; Hadfield, S.; Bowden, R. Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2020; pp. 10023–10033.
48. Zhao, X.; Wu, X.; Miao, J.; Chen, W.; Chen, P.C.Y.; Li, Z. ALIKE: Accurate and Lightweight Keypoint Detection and Descriptor Extraction. *IEEE Trans Multimedia* 2023, *25*, 3101–3112, doi:10.1109/TMM.2022.3155927.
49. Ismail, L.; Materwala, H.; P. Karduck, A.; Adem, A. Requirements of Health Data Management Systems for Biomedical Care and Research: Scoping Review. *J Med Internet Res* 2020, *22*, doi:10.2196/17508.
50. Ismail, L.; Materwala, H. A Review of Blockchain Architecture and Consensus Protocols: Use Cases, Challenges, and Solutions. *MDPI Symmetry* 2019, *11*, 1198.
51. Hennebelle, A.; Ismail, L.; Materwala, H.; Al Kaabi, J.; Ranjan, P.; Janardhanan, R. Secure and Privacy-Preserving Automated Machine Learning Operations into End-to-End Integrated IoT-Edge-Artificial Intelligence-Blockchain Monitoring System for Diabetes Mellitus Prediction. *Comput Struct Biotechnol J* 2024, *23*, 212–233, doi:10.1016/j.csbj.2023.11.038.
52. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.-J. BLEU: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02; Association for Computational Linguistics: Morristown, NJ, USA, 2001; p. 311.
53. Lin, C.-Y. Rouge: A Package for Automatic Evaluation of Summaries. In Proceedings of the Text summarization branches out; 2004; pp. 74–81.
54. Google AI MediaPipe Solutions Guide Available online: https://ai.google.dev/edge/mediapipe/solutions/guide (accessed on 20 November 2024).
55. Li, S.; Jin, X.; Xuan, Y.; Zhou, X.; Chen, W.; Wang, Y.-X.; Yan, X. Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting. In Proceedings of the Advances in neural information processing systems; 2019.
56. Chattopadhyay, R.; Tham, C.-K. A Position Aware Transformer Architecture for Traffic State Forecasting. In Proceedings of the IEEE 99th Vehicular Technology Conference; IEEE: Singapore, July 26 2024.
57. Shahin, N.; Ismail, L. GLoT: A Novel Gated-Logarithmic Transformer for Efficient Sign Language Translation. In Proceedings of the 2024 IEEE Future Networks World Forum (FNWF); IEEE, October 15 2024; pp. 885–890.

 doi:10.20944/preprints202510.0130.v1

58. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In Proceedings of the Proceedings of the 23rd international conference on Machine learning - ICML '06; ACM Press: New York, New York, USA, 2006; pp. 369–376.
59. Heafield, K. KenLM: Faster and Smaller Language Model Queries. In Proceedings of the Proceedings of the sixth workshop on statistical machine translation; 2011; pp. 187–197.
60. Kudo, T.; Richardson, J. SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing. In Proceedings of the Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 66–71.
61. Sennrich, R.; Haddow, B.; Birch, A. Neural Machine Translation of Rare Words with Subword Units. In Proceedings of the Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 1715–1725.
62. Ma, H.; Li, Q.; Zheng, Y.; Zhang, Z.; Liu, X.; Gao, Y.; Al-Sarawi, S.F.; Abbott, D. MUD-PQFed: Towards Malicious User Detection on Model Corruption in Privacy-Preserving Quantized Federated Learning. *Comput Secur* 2023, *133*, 103406, doi:10.1016/j.cose.2023.103406.
63. Akhtar, N.; Saddique, M.; Asghar, K.; Bajwa, U.I.; Hussain, M.; Habib, Z. Digital Video Tampering Detection and Localization: Review, Representations, Challenges and Algorithm. *Mathematics* 2022, *10*, 168, doi:10.3390/math10020168.
64. Ghimire, S.; Choi, J.Y.; Lee, B. Using Blockchain for Improved Video Integrity Verification. *IEEE Trans Multimedia* 2020, *22*, 108–121, doi:10.1109/TMM.2019.2925961.
65. Sasikumar, K.; Nagarajan, S. Enhancing Cloud Security: A Multi-Factor Authentication and Adaptive Cryptography Approach Using Machine Learning Techniques. *IEEE Open Journal of the Computer Society* 2025, *6*, 392–402, doi:10.1109/OJCS.2025.3538557.