Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.078727

ARTICLE

Gloss-Internal Graph Construction and Encoding for Sign Language Translation

Sam Nguyen-Xuan[1,*] **and Han Nguyen**[2]

[1]Department of Computer Science, Swinburne Vietnam, FPT University, Ho Chi Minh City, Vietnam
[2]Bellini College of Artificial Intelligence, Cybersecurity and Computing, University of South Florida, Tampa, FL, USA

*Corresponding Author: Sam Nguyen-Xuan. Email: samnx2@fe.edu.vn

Received: 06 January 2026; Accepted: 13 March 2026; Published: 08 May 2026

ABSTRACT: We propose a Gloss-Internal Graph Construction and Encoding framework that represents compound glosses as directed, labeled graphs and integrates them into a Transformer via a graph-aware encoder. We evaluate our approach against Rule-Based Gloss Decomposition (RBGD) and Linear Gloss Sequence Encoding (LGSE) baselines on ASLG-PC12 and PHOENIX-2014T. Results show consistent improvements over both baselines, achieving gains of up to +3.2 BLEU-4 over LGSE and +7.0 BLEU-4 over RBGD on ASLG-PC12. On PHOENIX-2014T, our method yields gains of up to 1.9 BLEU-4 on the development set and 2.4 BLEU-4 on the test set. Ablation studies further indicate that agreement and reference edges contribute most to translation quality, that attention pooling outperforms mean pooling for graph-level aggregation, and that a single message-passing step offers a reasonable accuracy–efficiency trade-off for the compact gloss-internal graphs encountered in practice. These results suggest that explicit modeling of gloss-internal structure is a promising direction for sign language translation.

KEYWORDS: Sign language translation; gloss-to-Text translation; gloss-internal graph; sign language gloss; transformer-based models

1 Introduction

Sign language translation (SLT) seeks to convert sign language expressions into fluent and grammatically well-formed natural language sentences, serving as an essential assistive technology for deaf and hard-of-hearing communities. The most widely adopted SLT frameworks [1,2] formulate the problem as a two-stage pipeline. The first stage, Sign-to-Gloss [3–6], maps visual sign inputs into symbolic gloss sequences that abstract core linguistic content. The second stage, Gloss-to-Text, translates these intermediate gloss representations into written language. Despite advances in visual feature learning and end-to-end training, Gloss-to-Text remains a persistent bottleneck due to the distinctive linguistic structure and annotation conventions of sign language glosses [7,8]. Several studies have examined the limitations of gloss-based pipelines, noting that gloss annotations often provide only a coarse symbolic representation of the underlying signing signal and may introduce information loss during the annotation process [9].

Existing approaches [10–12] indicate that many limitations in SLT arise not only from model capacity but also from data and representation. Restricted dataset accessibility, heterogeneous annotation practices, and inconsistent glossing conventions impede reproducibility and hinder cross-dataset generalization. Consequently, gains achieved by powerful neural architectures often fail to transfer across corpora or linguistic settings. This observation suggests that the bottleneck lies as much in how sign language data are represented as in the capacity of state-of-the-art models to exploit those representations effectively.

Recent work has also explored improved modeling strategies for SLT. Chen et al. [13] introduce multimodal transfer learning to improve cross-modal alignment between visual features and textual representations. Subsequent studies investigate richer contextual and semantic modeling of gloss sequences, including semantic-aware gloss representations [14], contextual translation architectures [15,16], and large-scale multimodal or language-model-based approaches [17–21]. While these approaches improve model capacity and contextual reasoning, most systems still represent gloss inputs as linear token sequences. Consequently, the internal linguistic structure of compound glosses remains largely implicit within current modeling pipelines.

At the core of this representational challenge is the nature of gloss tokens themselves. Glosses [22] are not simple lexical units. They frequently compress multiple linguistic functions into a single symbol, including pronominal reference, person agreement, tense, aspect, and lexical meaning. For example, the compound gloss `IX-2P-WAKE-UP` simultaneously encodes indexical pointing, second-person agreement, and a verbal predicate. This structural compression makes gloss tokens highly informative, but it also renders their internal linguistic organization essential for accurate translation. However, this organization remains implicit under conventional rule-based and linear gloss representations.

Rule-Based Gloss Decomposition (RBGD) applies deterministic linguistic or heuristic rules to split compound glosses into smaller functional units prior to translation [23,24]. For example, the compound gloss `IX-2P-WAKE-UP` is decomposed into `[IX]`, `[2P]`, `[WAKE]`, and `[UP]`, corresponding to an indexical reference marker, a second-person agreement marker, a verbal root, and an aspectual modifier. The decomposed units are then fed into a Transformer encoder–decoder as a flat token sequence. This approach reduces vocabulary sparsity and exposes meaningful subcomponents, but the resulting representation remains linear: the structural dependencies among decomposed units, such as the agreement relation between `2P` and `WAKE` or the modification relation between `UP` and `WAKE`, are not explicitly encoded and must be implicitly inferred by the sequence model.

Linear Gloss Sequence Encoding (LGSE) represents each gloss token as an element of a flat sequence without prior decomposition [1–3,13]. Each gloss is treated as either an atomic symbol or a subword unit produced by statistical tokenization methods such as WordPiece or SentencePiece [25,26]. Applied to `IX-2P-WAKE-UP`, this produces fragments such as `[IX]`, `[-]`, `[2]`, `[P]`, `[WAKE]`, and `[UP]`, which are encoded as a flat sequence `[IX]` → `[-]` → `[2]` → `[P]` → `[WAKE]` → `[UP]`. While subword segmentation improves vocabulary reuse and mitigates sparsity, the resulting token stream remains agnostic to the linguistic roles and grammatical dependencies embedded within compound glosses.

Graph-based representations for sign language processing have primarily been explored at the visual level, where spatio-temporal graphs are constructed over body joints or keypoints to model sign articulation and motion dynamics [27]. More broadly, graph neural networks and graph transformer architectures provide a general framework for learning over structured relational data [28].

To address this limitation, we propose Gloss-Internal Graph Representation (GIGR) as a structured alternative to rule-based decomposition and linear gloss-sequence representations for Gloss-to-Text translation. Rather than treating each gloss as an atomic token or a flat sequence of subunits, our approach represents compound glosses as directed, labeled graphs. Nodes correspond to linguistically grounded units (e.g., lexical roots, agreement markers, and aspectual modifiers), while edges encode their semantic and grammatical relations. This representation preserves gloss-internal dependencies and enables composition that is not restricted to surface token order. We operationalize this idea by introducing a *GIGR* module between the *Raw Gloss Text layer* and an *encoder–decoder Transformer layer*. Importantly, our method does not require modifying the downstream Transformer architecture, thereby preserving the efficiency and modularity of Transformer-based models. Our contributions are summarized as follows:

- We propose a *GIGR* module that converts compound glosses into directed, labeled graphs, capturing semantic and grammatical relations within gloss tokens that are not captured by flat token order.
- We propose a finite-state machine for constructing gloss-internal graphs, ensuring well-formedness, interpretability, and compatibility with established gloss annotation conventions.
- We design a graph-aware encoding module that converts gloss-internal graphs into compact, structure-aware embeddings, enabling seamless integration with standard Transformer encoder–decoder architectures without architectural modification.
- We demonstrate on ASLG-PC12 and PHOENIX-2014T that the proposed approach consistently improves translation quality over strong linear and rule-based baselines, while maintaining competitive end-to-end inference efficiency.

The remainder of this paper is organized as follows. Section 2 reviews related work on Gloss-to-Text translation and SLT. Section 3 details the proposed approach. Section 4 describes the experimental design. Section 5 presents the experimental results and discussion. Finally, Section 6 concludes the paper and future research directions.

2 Related Work

In this section, we categorize prior Gloss-to-Text translation approaches according to how gloss representations are modeled, rather than by the choice of neural architecture. We also review commonly used gloss datasets and analyze how compound glosses are represented within these annotation schemes.

2.1 Gloss-to-Text Translation

Gloss-to-Text translation aims to convert sign language gloss sequences into grammatically well-formed natural language sentences. In most existing SLT frameworks, this task is formulated as a neural sequence-to-sequence problem [29–32] and implemented using Transformer encoder–decoder architectures [33]. These models have demonstrated strong performance on standard benchmarks for gloss-based translation.

The most widely used evaluation dataset is PHOENIX-2014T [1], which contains German Sign Language gloss annotations aligned with German translations and serves as a standard benchmark for many SLT and Gloss-to-Text studies. Another commonly used dataset is ASLG-PC12 [22], which provides American Sign Language gloss sequences paired with English sentences and has been adopted in several recent Gloss-to-Text studies to investigate translation modeling and data scaling effects [3,7,13].

Recent research has explored improved modeling strategies for Gloss-to-Text translation. Chen et al. [13] propose a multimodal transfer learning framework that enhances alignment between visual representations and textual outputs. Other studies focus on richer contextual modeling of gloss sequences, including semantic-aware gloss representations [14], context-enhanced translation architectures [15,16], and large-scale multimodal or language-model-driven approaches [17–21]. These approaches improve contextual reasoning and translation quality by leveraging stronger encoders and pretrained language models.

Despite these advances, most existing methods treat gloss sequences as linear token streams in which each gloss is represented as an atomic symbol. However, gloss vocabularies in real datasets are typically large and sparse, and many gloss tokens encode compound forms that combine lexical roots with grammatical markers such as agreement, aspect, or referential indexing. These compound gloss forms are commonly expressed through annotation conventions such as hyphenation, suffix markers, or token concatenation.

Although such conventions preserve linguistic information for human annotators, their internal structure is rarely represented explicitly in machine-readable form. Consequently, the compositional structure embedded within compound gloss tokens remains largely implicit in existing Gloss-to-Text pipelines.

Recent work has attempted to reduce reliance on manual gloss annotations through pseudo-gloss generation for SLT [10]. However, pseudo-gloss sequences are also typically modeled as linear token streams, limiting the ability of neural translation models to capture the structural relationships within compound gloss expressions.

2.2 Compound Gloss Decomposition

Most Gloss-to-Text translation systems adopt LGSE [1–3,13], in which a signed utterance is represented as a flat sequence of gloss tokens that are processed by neural sequence-to-sequence translation models. In this paradigm, each gloss is treated as either an atomic token or a sequence of sub-units, and the linguistic structure of the utterance is assumed to be recoverable from token order and contextual co-occurrence. However, this assumption becomes problematic for sign language glosses, where individual gloss tokens often encode internal morphological or functional structure that is not explicitly represented in the linear sequence.

To mitigate vocabulary sparsity in LGSE systems, subword tokenization methods such as WordPiece and SentencePiece are commonly applied [25,26,34]. These approaches segment rare or unseen gloss tokens into statistically frequent units, improving vocabulary reuse and model robustness. Recent Transformer-based Gloss-to-Text models have adopted subword segmentation to better handle the large gloss vocabularies observed in real datasets [7]. However, subword tokenization still produces flat token streams and does not explicitly encode the relations among gloss components, such as agreement, aspect, or referential markers.

Alternative LGSE variants operate at different granularity levels. Word-level encoding treats each gloss as an indivisible symbol, preserving gloss boundaries but suffering from severe sparsity due to the large number of compound gloss forms. Character-level encoding alleviates out-of-vocabulary issues but significantly increases sequence length and often weakens the model's ability to capture higher-level linguistic structure. In this work, we adopt a WordPiece-based LGSE implementation as a representative subword baseline.

A complementary line of research explores RBGD, which applies hand-crafted linguistic or heuristic rules to split compound glosses into smaller units before translation. Such approaches exploit consistent glossing conventions such as hyphenation, affixation, or functional marker concatenation to reduce lexical sparsity. For example, Koller et al. [27] employ deterministic decomposition rules to alleviate vocabulary sparsity in gloss-based modeling, while Forster et al. [35] incorporate rule-based normalization and decomposition within the RWTH Aachen SLT pipeline. Although these methods reduce vocabulary size, the resulting representations remain linear token sequences, and the structural dependencies among decomposed components are not explicitly modeled.

More broadly, linguistically informed gloss annotation schemes have been proposed to enrich gloss representations with grammatical information. Nguyen-Xuan et al. [8], for instance, incorporate agreement and reference information directly into gloss annotations to improve annotation consistency. Nevertheless, such enriched gloss tokens are still consumed by neural translation models as linear sequences, leaving gloss-internal linguistic relations implicit.

Existing approaches to gloss decomposition primarily focus on reducing vocabulary sparsity while maintaining a sequential representation. Explicit modeling of the structural relationships within compound gloss tokens remains largely unexplored, limiting the ability of neural translation models to capture the compositional linguistic structure embedded in gloss expressions.

2.3 Graph Representation Learning

Graph-based neural models have been widely applied in natural language processing to capture structured linguistic relations that extend beyond linear token sequences. In these models, linguistic units are represented as nodes and their dependencies as edges, allowing neural architectures to reason over relational structures. Early graph neural network approaches, such as Graph Convolutional Networks [36], and Graph Attention Networks [37], demonstrated the effectiveness of graph-based representations for modeling structured data.

More recent architectures extend the Transformer attention mechanism to graph-structured inputs. Graph transformer models enable global message passing while preserving relational topology, allowing the model to capture long-range dependencies in structured data [28,38].

In NLP, graph representations are commonly used to encode syntactic and semantic dependencies, improving tasks such as relation extraction, semantic role labeling, and machine translation [39–41]. These studies demonstrate that explicitly modeling linguistic relations helps neural models capture long-range dependencies that are difficult to infer from sequential token representations alone.

Sign language gloss annotations introduce additional structural complexity. Gloss sequences often contain explicit referential and predicate relations that are not fully captured by a simple linear token stream. For instance, the gloss sequence `IX-2P-WAKE-UP` denotes a second-person plural referent followed by the predicate `WAKE-UP`. Although the sequence appears linear, it implicitly encodes a subject–predicate relation in which the referential index `IX-2P` functions as the subject of the predicate `WAKE-UP`. In a purely sequential representation, this dependency remains implicit and must be inferred from positional context.

However, most existing Gloss-to-Text translation pipelines represent gloss annotations as flat token sequences, treating each gloss as an independent unit. Such representations do not explicitly capture the relational structure between referential indices and predicates, which may limit the model's ability to recover predicate–argument relations during translation. To address this limitation, we introduce the GIGR framework that models gloss sequences as structured graphs.

3 Proposed Approach

3.1 Problem Formulation

Let $G = (g_1, g_2, \ldots, g_n)$ denote an input sequence of sign language glosses, where each g_i is a symbolic gloss token, and let $Y = (y_1, y_2, \ldots, y_m)$ denote the corresponding natural language sentence. The goal of Gloss-to-Text translation is to learn a conditional mapping $P(Y \mid G)$ that generates a grammatically correct and semantically faithful target sentence.

Conventional Gloss-to-Text systems represent G as a linear token sequence and rely on neural sequence models to implicitly recover linguistic structure. However, as discussed in Section 2, individual gloss tokens often encode multiple interacting linguistic functions, making linear representations insufficient for capturing internal dependencies. To address this limitation, we propose a *graph-structured representation* that explicitly models the internal organization of glosses.

Fig. 1 illustrates the proposed GIGR framework for Gloss-to-Text translation. The framework transforms an input gloss sequence into a structured graph representation before feeding it into an encoder–decoder Transformer. Within GIGR, each compound gloss is modeled as a directed, labeled graph in which nodes represent linguistically grounded components such as lexical roots, referential indices, agreement markers, and aspectual modifiers, while edges encode the grammatical and semantic relations among these components.

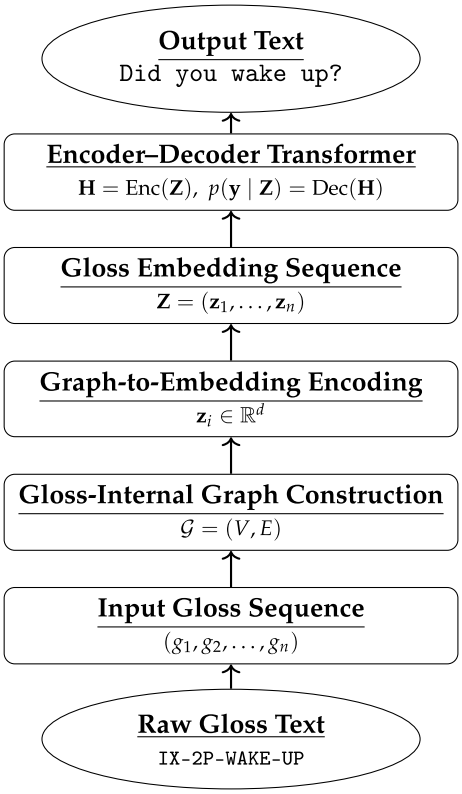


Figure 1: Proposed framework for gloss-to-text translation.

3.2 Input Gloss Sequence

The input gloss sequence G is obtained by tokenizing the raw gloss text into an ordered sequence of symbolic gloss units,

$$G = (g_1, g_2, \ldots, g_n), \tag{1}$$

where each g_i denotes an individual gloss annotated according to standard sign language glossing conventions. Tokenization preserves the original temporal order of the signed utterance.

Although the sequence G is linear at the utterance level, each gloss token g_i may internally encode multiple interacting linguistic functions. To recover this internal structure, each gloss token is processed independently by a deterministic FST, which enforces ordering and attachment constraints over functional components such as Reference, Agreement, Aspect, and Lexicalroot. This finite-state formulation augments the linear representation with structured, rule-governed internal dependencies, which are subsequently encoded as gloss-internal graphs.

Table 1 summarizes the rule inventory used for gloss decomposition. The same rule set is also compatible with the annotation conventions of PHOENIX-2014T glosses, where compound glosses are commonly formed by concatenating functional markers and lexical predicates using hyphens.

Among these categories, Lexicalroot, Reference, and Agreement form the core subset that accounts for most of the observed performance gains. Other categories, such as Aspect, Negation, and Locative, act as refinements that improve robustness and interpretability but are not strictly required for the method to function.

Table 1: FST transition rules for morphological analysis.

Component Pattern c	Role $\rho(c)$	Relation $\eta(c)$
IX-*	Reference	Reference
1P/2P/3P	Agreement	Agreement
NOT/NO	Negation	Modification
LOC/LOCATION	Locative	Modification
Aspect markers	Aspect	Modification
	Refinement (non-root) relation	
Reference–Agreement	—	`rel_agreement`
otherwise	Lexicalroot	—

Formally, the gloss-internal parsing process is modeled as an FST $\mathcal{M} = (\Sigma, Q, \delta, q_0, F)$, which maps input gloss strings to structured outputs.

$$\text{FST} : \Sigma^* \to (\Sigma \times \mathcal{R})^*, \tag{2}$$

where the input is a gloss $g \in \Sigma^*$ (e.g., `IX-2P-WAKE-UP`), and the output is a sequence of *(morpheme, role)* pairs, such as [(`IX`, Reference), (`2P`, Agreement), (`WAKE`, Lexicalroot), (`UP`, Aspect)].

The state set Q corresponds to linguistically grounded functional roles defined by the gloss annotation scheme. The transition function $\delta : Q \times \Sigma \to Q$ is deterministic and governed by the role-mapping rules summarized in Table 1. For a component $c \in \Sigma$, the rule table specifies (i) its assigned role $\rho(c) \in Q$ and (ii) its attachment relation $\eta(c)$. Together, these determine the valid state transition $\delta(q, c) = \rho(c)$.

Root-attached rules induce transitions that lead to the Lexicalroot state, while refinement rules define admissible transitions between non-root states. These constraints ensure that only well-formed sequences of functional components are accepted and that each compound gloss is assigned a unique, interpretable internal structure. Fig. 2 provides a graphical realization of the FST defined above, where nodes correspond to states in Q and directed edges represent admissible transitions specified by δ. Each path from Start to End encodes a valid ordering of functional components within a compound gloss.

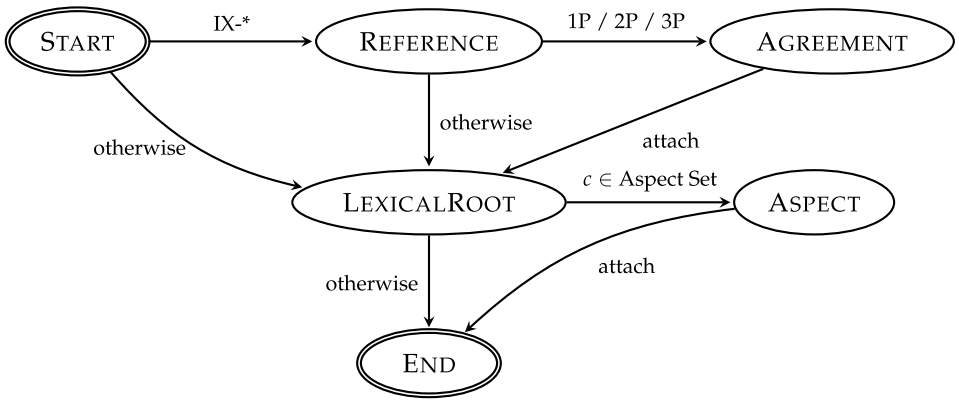


Figure 2: FST for deterministic ordering constraints over components of a single compound gloss.

The FST formulation provides several advantages. First, it guarantees $O(n)$ time complexity, where n is the number of morphemic components within a compound gloss, with deterministic parsing ensured by

construction. Second, the explicit state space and transition function offer full transparency and auditability, allowing each parsing decision to be traced back to linguistically interpretable rules. Third, the FST transitions encode well-established properties of sign language morphology, including the precedence of referential markers, the dependency of agreement morphology on prior reference establishment, and the post-verbal realization of aspectual markers. This linguistic grounding distinguishes our approach from purely data-driven methods, which typically require costly and manually annotated morphological parses.

For example, the ASLG-PC12 gloss `IX-2P-WAKE-UP` is segmented into the components `IX`, `2P`, `WAKE`, and `UP`. According to Table 1, `IX` is assigned the role Reference and attached to the lexical root via a *reference* relation, while `2P` is assigned the role Agreement and attached via an *agreement* relation. Aspectual markers such as `UP` are assigned the role Aspect and linked to the root through a *modification* relation. The remaining component `WAKE`, which does not match any specialized pattern, is assigned the role LexicalRoot and selected as the root of the gloss-internal graph. When both reference and agreement components are present, an additional refinement edge labeled `rel_agreement` is introduced from the agreement node to the reference node, capturing the dependency between person agreement and its associated referential index.

3.3 Gloss-Internal Graph Construction

Algorithm 1 executes the FST constraints in Table 1 to produce the directed labeled dependency graph. Each step of the algorithm corresponds to a specific aspect of the FST semantics.

Algorithm 1: Gloss-internal graph construction

Input: Compound gloss token g
Output: Directed labeled graph $\mathcal{G} = (V, E)$
Segmentation: split g at hyphen boundaries $C = (c_1, \ldots, c_n)$;
Initialize nodes: Set $V \leftarrow \{1, \ldots, n\}$ and $E \leftarrow \varnothing$;
if $n = 1$ **then**
    Assign role $\rho(c_1) \leftarrow$ `LexicalRoot`;
    return $\mathcal{G} = (V, E)$;
Role assignment: For each $c_i \in C$, assign role $\rho(c_i)$ using Table 1;
Root selection: Let k be the first index such that $\rho(c_k) =$ `LexicalRoot`;
if *no such index exists* **then**
    set $k \leftarrow n$;
Root-attached edges: foreach $i \in \{1, \ldots, n\} \backslash \{k\}$ **do**
    Assign relation label $\eta(c_i)$ using Table 1;
    Add directed edge $(i \rightarrow k,\ \eta(c_i))$ to E;
Agreement refinement: if *there exist indices (i, j) such that* $\rho(c_i) =$ `REFERENCE` *and* $\rho(c_j) =$ `AGREEMENT` **then**
    Add refinement edge $(j \rightarrow i,$ `rel_agreement`$)$ to E;
return $\mathcal{G} = (V, E)$;

The algorithm begins by segmenting the input gloss token g at hyphen boundaries, yielding an ordered sequence of components $C = (c_1, \ldots, c_n)$. A node is created for each component, and the node set is initialized as $V = \{1, \ldots, n\}$, preserving the original left-to-right order of the gloss components. If the gloss consists of a single component ($n = 1$), it is assigned the role LexicalRoot and returned as a trivial single-node graph.

For multi-component glosses, each component c_i is assigned a functional role $\rho(c_i) \in Q$ using the deterministic pattern rules in Table 1. This role assignment step corresponds to realizing the FST state reached after consuming component c_i. At this stage, the FST constrains which role assignments are admissible, but no graph edges are yet constructed.

Next, the algorithm selects a unique root node by identifying the first component whose role is LEXICALROOT. This enforces the FST constraint that every valid compound gloss must contain a lexical predicate that anchors the internal structure. If no such component is identified, the final component is selected as a fallback root to ensure graph well-formedness.

For each non-root component c_i, the algorithm assigns a relation label $\eta(c_i)$ according to Table 1 and adds a directed edge $(i \to k,\ \eta(c_i))$ to the edge set E, where k denotes the root index. These edges correspond to FST-sanctioned root attachments, converting admissible FST transitions into explicit dependency relations in the gloss-internal graph.

Finally, if both a REFERENCE component and an AGREEMENT component are present, the algorithm introduces an additional refinement edge $(j \to i,\ \texttt{rel\_agreement})$, where $\rho(c_j) =$ AGREEMENT and $\rho(c_i) =$ REFERENCE. This refinement step captures a non-root dependency permitted by the FST, reflecting the fact that agreement markers modify an existing referential index rather than attaching directly to the lexical predicate.

The algorithm returns the directed labeled graph $\mathcal{G} = (V, E)$, which encodes both the FST-constrained ordering of functional roles and the resulting gloss-internal dependency structure.

3.4 Graph-to-Embedding Encoding

Given the gloss-internal graph $\mathcal{G} = (V, E)$, the objective of graph-to-embedding encoding is to map the structured representation into a fixed-dimensional vector $\mathbf{z}_i \in \mathbb{R}^d$ that summarizes the linguistic content of the gloss token g_i.

Each node $v \in V$ is first mapped to an initial embedding $\mathbf{h}_v^{(0)}$ according to its lexical form and role type. To incorporate relational information, node representations are updated using relation-aware aggregation over incoming edges:

$$\mathbf{h}_v^{(1)} = \phi\left(\mathbf{h}_v^{(0)}, \sum_{(u,v,r)\in E} \psi_r\left(\mathbf{h}_u^{(0)}\right)\right), \tag{3}$$

where $\psi_r(\cdot)$ is a relation-specific transformation and $\phi(\cdot)$ denotes a non-linear update function. This operation allows each node to integrate contextual information from linguistically related units while preserving edge semantics.

The final gloss embedding is obtained by aggregating the updated node representations:

$$\mathbf{z}_i = \text{POOL}\left(\left\{\mathbf{h}_v^{(1)} \mid v \in V_i\right\}\right), \tag{4}$$

where $\texttt{POOL}(\cdot)$ denotes a permutation-invariant aggregation operator. We adopt attention-based pooling $\mathbf{z}_i = \sum_{v\in V_i} \alpha_v \mathbf{h}_v^{(1)}$. The resulting vector $\mathbf{z}_i$ serves as a compact and structure-aware representation of the gloss token, forming the interface between gloss-internal graph modeling and downstream sequence-level translation.

3.5 Gloss Embedding Sequence

After graph-to-embedding encoding, each gloss token g_i is represented by a fixed-dimensional vector $\mathbf{z}_i \in \mathbb{R}^d$. Given an input gloss sequence $G = (g_1, g_2, \ldots, g_n)$, the corresponding gloss embedding sequence is formed as

$$\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n). \tag{5}$$

This sequence preserves the original temporal order of gloss tokens while embedding gloss-internal structural information within each vector $\mathbf{z}_i$. In contrast to raw gloss sequences, the embedding sequence provides a continuous, structure-aware representation that is directly compatible with sequence-based neural architectures.

Although the rule inventory is linguistically motivated, it is not assumed to be exhaustive. Glosses that do not match any predefined pattern are treated as single-node graphs in which the full token itself is assigned the role LEXICALROOT, ensuring well-formedness for all inputs regardless of annotation variation or rare compound forms. The robustness of this fallback strategy is evaluated empirically in Section 5.3, where the model is shown to degrade gracefully under partial rule coverage. The resulting gloss embedding sequence $\mathbf{Z}$ serves as the input to the subsequent Transformer-based translation module.

4 Experimental Design

4.1 Experimental Setup

To evaluate our proposed method systematically, we design an experimental setup that ensures fair and reproducible model comparison. All experiments are conducted on an NVIDIA GPU equipped with 16 GB of memory. The models are implemented in Python 3.10 using PyTorch 2.1 and the HuggingFace Transformers library, with CUDA 11.8 for GPU acceleration. A detailed summary of the experimental setup and hyperparameters is provided in Table 2.

Table 2: Selected parameters.

Hyperparameter	Value
Optimizer	AdamW
Batch size	32
Learning rate	5×10^{-5}
Epochs	10
Dropout	0.2
Number of attention heads	8
Encoder layers	2/4/6
Decoder layers	2/4/6
Word embedding size	512
GPU	NVIDIA, 16 GB memory

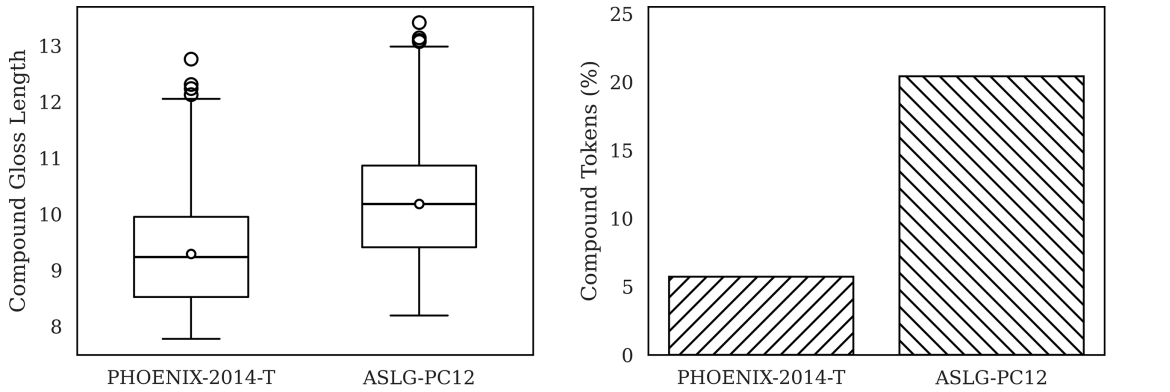
We employ the AdamW optimizer with a batch size of 32. The embedding dimension is fixed at 512, and the hidden dimension of the position-wise feed-forward networks is set to 2048. Multi-head self-attention employs 8 attention heads in all configurations. All models are trained for 10 epochs with an initial learning rate of 5×10^{-5}.

We briefly describe the baseline models employed in the experimental evaluation. To ensure a fair and controlled comparison, all baselines share the same encoder–decoder architecture, training procedure, hyperparameter configuration, and implementation framework. We also vary the number of layers in both the encoder and decoder among $\{2, 4, 6\}$, consistent with established Transformer-based architectures. The evaluated baseline models are summarized as follows:

- **LGSE** represents gloss inputs as flat token sequences and encodes them using word-level or subword-level representations, following conventional neural machine translation paradigms [1,2,13].
- **RBGD** applies deterministic linguistic or heuristic rules to split compound glosses into smaller functional units prior to translation [23,27].

4.2 Benchmark Datasets

To better characterize the structural properties of the benchmark datasets, we analyze the prevalence and form of compound glosses. Fig. 3 reports statistics for PHOENIX-2014T and ASLG-PC12. Compound glosses account for 5.7% of tokens in PHOENIX-2014T and 20.4% in ASLG-PC12, indicating substantial variation in structural density across datasets. In PHOENIX-2014T, compound glosses are formed almost exclusively through hyphen-based composition, which explicitly marks internal boundaries between functional components. In contrast, ASLG-PC12 exhibits a higher proportion of plus-based compounding, resulting in longer and more structurally opaque gloss tokens. These differences motivate explicit modeling of gloss-internal relations, particularly for datasets such as ASLG-PC12 where compound structures are frequent and internal boundaries are less explicit. For PHOENIX-2014T, we follow the standard dataset split. For ASLG-PC12, we apply a 70/15/15 stratified split.



(a) Distribution of compound gloss lengths. (b) Percentage of compound gloss tokens per dataset.

Figure 3: Compound gloss statistics on benchmark datasets: (**a**) distribution of compound gloss lengths and (**b**) percentage of compound gloss tokens per dataset.

4.3 Metrics

4.3.1 Bilingual Evaluation Understudy

Bilingual Evaluation Understudy (BLEU) [42,43] is a widely used automatic metric for evaluating machine translation quality. BLEU measures the overlap between candidate translations and reference translations using modified n-gram precision and a brevity penalty to discourage overly short outputs. Formally, BLEU is defined as

$$\text{BLEU} = \text{BP} \cdot \exp\left(\frac{1}{N}\sum_{n=1}^{N}\log p_n\right), \tag{6}$$

where p_n denotes the modified precision of n-grams and N is the maximum n-gram order (typically $N = 4$ for BLEU-4). The brevity penalty BP is defined as

$$\mathrm{BP} = \begin{cases} 1, & c > r, \\ \exp\left(1 - \frac{r}{c}\right), & c \le r, \end{cases} \tag{7}$$

where c and r denote the lengths of the candidate and reference translations, respectively. In this work, we report BLEU-1 through BLEU-4 at the corpus level.

4.3.2 chrF

We additionally report chrF [44], a character n-gram F-score that measures overlap between generated and reference texts at the character level. Compared with word-level metrics, chrF is more robust to tokenization differences and morphological variation, making it particularly suitable for Gloss-to-Text translation, especially for morphologically rich languages such as German.

4.3.3 Evaluation Configuration

All BLEU and chrF scores are computed using the `sacreBLEU` toolkit (v2.3.1) [45], which provides standardized and reproducible evaluation independent of external tokenization tools.

For ASLG-PC12, BLEU is computed using the `13a` tokenizer with case-sensitive scoring. For PHOENIX-2014T, we use the `intl` tokenizer to better handle German morphology and compound nouns. No additional detokenization is applied beyond the internal preprocessing performed by `sacreBLEU`.

chrF scores are computed using the `sacreBLEU` implementation with character order $n = 6$ and word order $n = 2$, corresponding to the default chrF++ configuration. The complete evaluation scripts and configuration files are available in the public repository at https://github.com/TeddySNGUYEN/Gloss2Text.

4.3.4 End-to-End Latency

End-to-end latency measures the total inference time required to process an input and produce a final output. In our setting, this includes preprocessing, gloss-internal graph construction, encoder forward passes, and decoder generation. The latency T_{e2e} (ms/sample) is measured with batch size $B = 32$ and defined as

$$T_{\mathrm{e2e}} = T_{\mathrm{pre}} + T_{\mathrm{graph}} + T_{\mathrm{enc}} + T_{\mathrm{dec}}, \tag{8}$$

where T_{pre} denotes preprocessing time, T_{graph} the cost of graph construction, and T_{enc} and T_{dec} the encoder and decoder inference times, respectively. This metric reflects the practical deployment cost of the system and captures the cumulative overhead introduced by structural modeling.

5 Results and Discussion

5.1 Comparison of Translation Performance

Table 3 reports the performance comparison on ASLG-PC12 using BLEU-1 to BLEU-4 and chrF for both development and test sets. Overall, the proposed method (GIGR) consistently outperforms the two baselines (RBGD and LGSE) across all metrics, indicating that explicitly modeling gloss-internal structure yields more accurate and fluent Gloss-to-Text translations.

Table 3: Performance comparison on the ASLG-PC12 dataset.

Method	Dev Set					Test Set				
	BLEU-1	**BLEU-2**	**BLEU-3**	**BLEU-4**	**chrF**	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**chrF**
RBGD baseline	90.10	82.20	75.90	73.40	81.20	89.20	81.10	74.60	71.90	80.10
LGSE baseline	90.40	85.60	79.32	76.90	84.30	90.80	85.90	78.60	75.70	83.60
Our proposal (GIGR)	93.90	86.10	81.90	79.40	86.10	92.40	85.60	81.40	78.90	85.70

On the development set, GIGR achieves BLEU-4 of 79.40, improving over LGSE (76.90) by +2.50 BLEU-4 and over RBGD (73.40) by +6.00 BLEU-4. Similar trends are observed on the test set, where GIGR reaches a BLEU-4 score of 78.90, surpassing LGSE by +3.20 BLEU-4 and RBGD by +7.00 BLEU-4. The consistent gains across BLEU orders suggest that GIGR improves not only unigram-level lexical selection (BLEU-1), but also higher-order n-gram coherence (BLEU-3/4), reflecting better phrase formation and sentence-level fluency.

The improvements in chrF further support this conclusion. On the test set, GIGR increases chrF from 83.60 (LGSE) to 85.70, indicating better character-level matching and morphological consistency. This is particularly important for ASLG-PC12, where compound glosses and structured markers often correspond to subtle grammatical or functional variations in the target English output. Notably, the performance gap between GIGR and the baselines remains stable from the development set to the test set, indicating that the improvements generalize well to unseen data.

Notably, the advantage of GIGR over LGSE is larger for BLEU-4 than for BLEU-1, especially on the test set (+3.20 BLEU-4 vs. +1.60 BLEU-1). This indicates that gloss-internal graph modeling contributes more strongly to capturing longer-range dependencies and compositional structure, rather than merely improving local word choice. In contrast, the weaker RBGD baseline shows a substantial drop across BLEU-2 to BLEU-4, reflecting the limitation of purely rule-based decomposition, which reduces vocabulary sparsity but fails to preserve relational structure among functional gloss components.

Table 4 presents a performance comparison on the PHOENIX-2014T dataset, a challenging benchmark characterized by longer sentences, richer syntactic structure, and domain-specific weather terminology. Across both the development and test sets, the proposed method (GIGR) consistently outperforms the RBGD and LGSE baselines on all evaluation metrics, demonstrating the effectiveness of gloss-internal graph modeling under more complex translation conditions.

Table 4: Performance comparison on the PHOENIX-2014T.

Method	Dev Set					Test Set				
	BLEU-1	**BLEU-2**	**BLEU-3**	**BLEU-4**	**chrF**	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**chrF**
RBGD baseline	41.84	32.97	27.52	24.65	40.12	42.90	32.88	27.05	22.92	39.45
LGSE baseline	44.72	34.60	28.45	25.18	43.80	43.95	33.81	27.96	24.40	42.95
Our proposal (GIGR)	46.30	36.85	31.20	27.10	44.90	45.60	36.10	31.55	26.80	43.85

On the dev set, GIGR improves BLEU-4 from 25.18 (LGSE) to 27.10, corresponding to a gain of +1.92 BLEU-4, while also achieving the highest chrF score (44.90). Similar trends are observed on the test set, where GIGR attains a BLEU-4 score of 26.80, exceeding LGSE by +2.40 BLEU-4 and RBGD by +3.88 BLEU-4. Although these gains are smaller than those observed on ASLG-PC12, they remain consistent across all BLEU orders and chrF, indicating systematic benefits from incorporating explicit structural information into gloss representations.

The consistent improvements from BLEU-1 through BLEU-4 suggest that GIGR enhances both local lexical accuracy and higher-order n-gram coherence, while the chrF gains indicate better character-level fluency and improved morphological consistency in the generated German translations. Importantly, the performance advantage of GIGR remains stable from development to test sets, suggesting robust generalization rather than dataset-specific tuning.

Across both datasets, the proposed approach consistently outperforms strong baselines, confirming the effectiveness of gloss-internal graph modeling for Gloss-to-Text translation. On ASLG-PC12, GIGR achieves larger gains over LGSE (e.g., +3.2 BLEU-4 on the test set), which can be attributed to the higher prevalence and greater structural complexity of compound glosses in that corpus. In contrast, PHOENIX-2014T shows more moderate but reliable improvements, indicating that while explicit structural modeling generally benefits translation quality, its impact becomes more pronounced in settings with denser gloss-internal composition. Additional qualitative examples are provided in Appendix A.

5.2 End-to-End Latency Analysis

Fig. 4 shows that end-to-end inference latency increases sharply as encoder depth grows on both ASLG-PC12 and PHOENIX-2014T, while translation quality improves more gradually. Increasing the encoder from 2 to 4 layers raises latency from 22.02 to 35.47 ms on ASLG-PC12 and from 21.82 to 34.54 ms on PHOENIX-2014T. Further increasing the depth to 6 layers nearly triples the latency compared to the 2-layer configuration, reaching 58.60 and 57.63 ms, respectively. In contrast, BLEU-4 improves from 76.46 to 78.90 and 80.27 on ASLG-PC12, and from 23.76 to 24.40 and 26.80 on PHOENIX-2014T. These results indicate diminishing accuracy gains relative to the rapidly increasing computational cost, suggesting that a moderate encoder depth offers the most favorable trade-off between translation quality and inference efficiency.

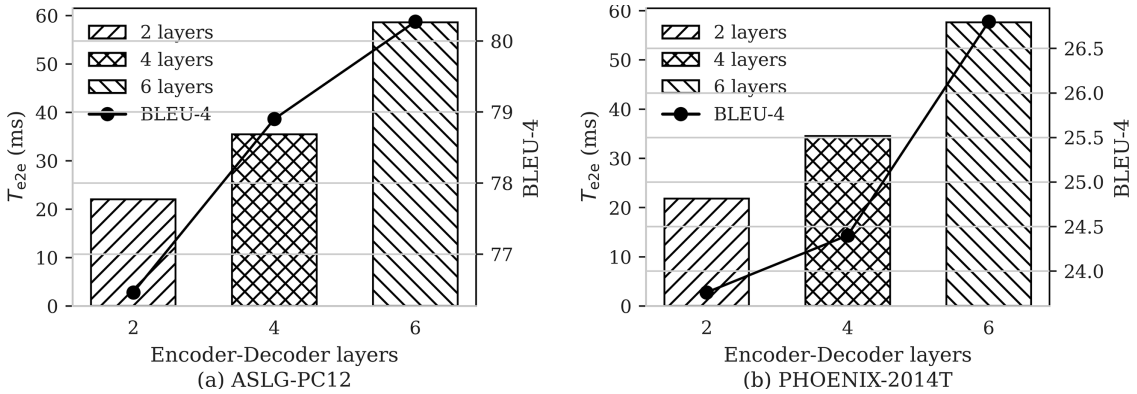


Figure 4: Effect of encoder depth on end-to-end latency: (**a**) ASLG-PC12 and (**b**) PHOENIX-2014T.

5.3 Ablation Study

5.3.1 Relation-Type Ablation

To better understand the contribution of different relation types in the proposed GIGR, we perform a relation-type ablation experiment. Specifically, we remove one category of relation during graph construction while keeping all other components unchanged. Table 5 reports the translation performance obtained under each configuration on ASLG-PC12 and PHOENIX-2014T.

Table 5: Relation-type ablation results.

Configuration	ASLG-PC12		PHOENIX-2014T	
	BLEU-4	chrF	BLEU-4	chrF
Full model	78.90	85.70	26.80	43.85
w/o agreement relation	72.10	82.50	23.00	41.90
w/o reference relation	73.40	83.10	23.40	41.30
w/o modifier relation	74.00	83.60	23.90	42.10

The results show that removing *agreement* or *reference* relations leads to the largest performance degradation across both datasets. This observation indicates that these relation types encode important grammatical and referential information within compound gloss tokens. In contrast, removing modifier relations results in a smaller performance drop, suggesting that modifier structures contribute useful but less critical information for gloss interpretation.

5.3.2 Robustness to Incomplete Rule Sets

To evaluate the robustness of the proposed graph construction mechanism, we simulate incomplete rule inventories by randomly disabling a portion of the decomposition rules during graph generation. Table 6 reports the translation performance when the available rule coverage is reduced from 100% to 80% and 60%.

The results indicate that translation performance gradually decreases as rule coverage is reduced. Nevertheless, the degradation remains moderate, demonstrating that the proposed graph encoder can still capture meaningful structural information even when the rule inventory is incomplete. This suggests that the method is robust to imperfect rule sets in practical sign language gloss annotations.

Table 6: Robustness to incomplete rule sets.

Rule Coverage	ASLG-PC12		PHOENIX-2014T	
	BLEU-4	chrF	BLEU-4	chrF
100% rules	78.90	85.70	26.80	43.80
80% rules	76.30	85.10	25.60	41.10
60% rules	74.20	83.40	24.10	39.60

5.3.3 Pooling Strategy Comparison

We further compare the proposed attention-based pooling mechanism with a simple mean pooling baseline when aggregating node representations into a graph-level embedding. Table 7 summarizes the results on both datasets.

Table 7: Pooling strategy comparison.

Pooling Method	ASLG-PC12		PHOENIX-2014T	
	BLEU-4	chrF	BLEU-4	chrF
Mean pooling	76.10	84.90	25.40	41.60
Attention pooling (ours)	78.90	85.70	26.80	43.85

Attention pooling consistently outperforms mean pooling across all evaluation metrics. These findings indicate that allowing the model to learn adaptive importance weights for different gloss components leads to more informative graph-level representations, thereby improving translation performance.

5.3.4 Message Passing Depth

Finally, we analyze the impact of the number of message-passing steps in the graph encoder. Increasing the number of propagation steps allows information to flow across a larger portion of the gloss graph, potentially improving the model's ability to capture structural dependencies.

Table 8 shows that increasing the depth beyond a single message-passing step yields only marginal improvements. Considering the additional computational cost introduced by deeper propagation, we adopt a single message-passing step as a practical trade-off between efficiency and translation performance.

Table 8: Effect of graph encoder depth.

Message Passing Steps	ASLG-PC12		PHOENIX-2014T	
	BLEU-4	chrF	BLEU-4	chrF
1 step (ours)	78.90	85.70	26.80	43.85
2 steps	79.10	86.90	26.90	41.62
3 steps	79.00	86.80	26.40	40.20

5.3.5 Parser Coverage Analysis

To assess the applicability of the proposed rule-based gloss parser, we report the proportion of gloss tokens that can be successfully parsed by the deterministic FST described in Section 3.2. A token is considered successfully parsed when at least one component matches the rule inventory and can be converted into a valid gloss-internal graph.

Table 9 summarizes the parsing coverage on the two benchmark datasets. The parser achieves coverage of **97.1%** on ASLG-PC12 and **94.3%** on PHOENIX-2014T, indicating that the rule inventory covers the vast majority of observed gloss tokens. Tokens that do not match any predefined pattern correspond mainly to rare gloss forms or annotation variations and are handled using the fallback strategy described in Section 3.5, where the entire gloss token is treated as a single-node graph with role LEXICALROOT.

Table 9: Parsing coverage of the gloss-internal graph constructor.

Dataset	Total Tokens	Parsed Tokens	Coverage (%)
ASLG-PC12	28,640	27,804	97.1
PHOENIX-2014T	58,420	55,091	94.3

5.4 Limitations and Future Directions

Despite its effectiveness, the proposed GIGR framework has several limitations. First, the gloss-internal graphs are constructed using hand-crafted, deterministic rules derived from glossing conventions. While this design provides interpretability and reduces data requirements, it may limit portability across datasets and sign languages with different annotation standards. In addition, incomplete rule coverage may lead to imperfect decomposition for rare or irregular compound gloss forms.

Second, the current graph construction focuses on modeling intra-gloss structure and does not explicitly capture higher-level discourse or inter-gloss dependencies beyond what is learned by the Transformer encoder. Extending the framework to incorporate sentence-level syntactic structure or cross-gloss relational information may further improve translation quality. Our evaluation is also restricted to Gloss-to-Text translation benchmarks, which do not include the visual modality.

6 Conclusion and Future Directions

This paper presents a GIGR framework for Gloss-to-Text sign language translation. By modeling compound glosses as directed graphs and encoding their internal linguistic structure, the proposed approach captures dependencies that are not explicitly represented in linear gloss sequences. Experimental results on ASLG-PC12 and PHOENIX-2014T demonstrate that incorporating gloss-internal relations consistently improves translation performance over conventional linear baselines.

Future work will investigate extending the framework to model cross-gloss dependencies and sentence-level structures, as well as exploring hybrid FST–neural approaches that combine rule-based interpretability with data-driven adaptation. Integrating the proposed representation into end-to-end sign language translation systems that jointly learn from visual sign inputs also constitutes another promising avenue.

Acknowledgement: The authors acknowledge institutional support from Swinburne Vietnam, FPT University.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The paper was primarily conceptualized and developed by Sam Nguyen-Xuan, who was responsible for the research design, methodology development, validation, and investigation. Han Nguyen contributed to the study by supporting data curation and simulation experiments. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Source code of our approach is available at: https://github.com/TeddySNGUYEN/Gloss2Text.

Ethics Approval: Not applicable.

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

Appendix A

This appendix provides additional qualitative translation examples to complement the quantitative results reported in the main paper. Tables A1 and A2 present representative outputs on the ASLG-PC12 and PHOENIX-2014T datasets, respectively.

Table A1: Qualitative translation examples on ASLG-PC12 produced by the proposed method.

Item	Text	Comment
Sample 1 (Correct)		
Gloss sequence	`BATH NEED YOU`	
Reference	You need to take a bath.	
Prediction	You need to take a bath.	Meaning preserved.
Sample 2 (Correct)		
Gloss sequence	`LONG SEE NO HOW YOU`	
Reference	Long time no see, how are you doing?	
Prediction	Long time no see, how are you doing?	Idiom preserved.
Sample 3 (Correct)		
Gloss sequence	`YOU WORK WHERE`	
Reference	Where do you work?	
Prediction	Where do you work?	Correct question structure.
Sample 4 (Correct)		
Gloss sequence	`YOUR NEW CAR COLOR WHAT`	
Reference	What color is your new car?	
Prediction	What color is your new car?	Correct wh-fronting.

Table A2: Qualitative translation examples on PHOENIX-2014T produced by the proposed method.

Item	Text	Comment
Sample 1 (Correct)		
Gloss sequence	REGEN SCHNEE REGION VERSCHWINDEN NORD REGEN KOENNEN REGION STERN KOENNEN SEHEN	
Reference	regen und schnee lassen an den alpen in der nacht nach im norden und nordosten fallen hier und da schauer sonst ist das klar	
Prediction	der regen fällt im norden und osten fällt heute nacht gebietsweise schnee sonst ist es meist klar	Key events preserved.
Sample 2 (Error)		
Gloss sequence	DONNERSTAG NORDWEST REGEN REGION SONNE WOLKE WECHSELHAFT DANN FREITAG AEHNLICH WETTER	
Reference	am donnerstag regen in der nordhälfte in der südhälfte mal sonne mal wolken ähnliches wetter dann auch am freitag	
Prediction	am donnerstag regnet es im norden und westen noch mal sonne mal wolken und es regnet es bleibt	Repetition; Friday omitted.
Sample 3 (Correct)		
Gloss sequence	KRAEFTIG AB MORGEN FRUEH MEISTENS SCHNEE SCHNEIEN KALT REGEN	
Reference	vom nordmeer zieht ein kräftiges tief heran und bringt uns ab den morgenstunden heftige schneefälle zum teil auch gefrierenden regen	
Prediction	und der wird morgen früh dann auch wieder schnee und es wird wieder schneien	Event and timing captured.
Sample 4 (Correct)		
Gloss sequence	WOCHENENDE SONNE SAMSTAG SCHOEN TEMPERATUR BIS SIEBZEHN GRAD REGION	
Reference	sonnig geht es auch ins wochenende samstag ein herrlicher tag mit temperaturen bis siebzehn grad hier im westen	
Prediction	am wochenende scheint verbreitet die sonne und es wird auch am samstag dann temperaturen bis siebzehn grad	Weekend and temperature preserved.

References

1. Camgoz N, Hadfield S, Koller O, Ney H, Bowden R. Neural sign language translation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–22; Salt Lake City, UT, USA. Piscataway, NJ, USA: IEEE; 2018. p. 7784–93. doi:10.1109/CVPR.2018.00812.
2. Camgoz N, Koller O, Hadfield S, Bowden R. Sign language transformers: joint end-to-end sign language recognition and translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020 Jun 13–19; Virtual. Piscataway, NJ, USA: IEEE; 2020. p. 10023–33. doi:10.1109/CVPR42600.2020.01004.
3. Yin K, Read J. Better sign language translation with STMC-transformer. In: Proceedings of the 28th International Conference on Computational Linguistics; 2020 Dec 8–13; Barcelona, Spain. Stroudsburg, PA, USA: International Committee on Computational Linguistics; 2020. p. 5975–89. doi:10.18653/v1/2020.coling-main.525.
4. De Coster M, Van Herreweghe M, Dambre J. Sign language recognition with transformer networks. In: Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020); 2020 May 11–16; Marseille, France. Stroudsburg, PA, USA: European Language Resources Association (ELRA); 2020. p. 6018–24.
5. Eunice J, Andrew J, Sei Y, Hemanth DJ. Sign2Pose: a pose-based approach for gloss prediction using a transformer model. Sensors. 2023;23(5):2853. doi:10.3390/s23052853.
6. Saunders B, Camgoz NC, Bowden R. Progressive transformers for end-to-end sign language production. In: Vedaldi A, Bischof H, Brox T, Frahm JM, editors. Proceedings of the Computer Vision—ECCV 2020. Lecture Notes in Computer Science; 2020 Aug 23–28; Glasgow, UK. Cham, Switzerland: Springer; 2020. p. 687–705. doi:10.1007/978-3-030-58621-8_40.
7. Fayyazsanavi P, Anastasopoulos A, Kosecka J. Gloss2Text: sign language gloss translation using LLMs and semantically aware label smoothing. In: Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024; 2024 Nov 12–16; Miami, FL, USA. Stroudsburg, PA, USA: Association for Computational Linguistics; 2024. p. 16162–71. doi:10.18653/v1/2024.findings-emnlp.947.
8. Nguyen-Xuan S, Le GD, Nguyen H. A gloss annotation scheme for enhanced sign language translation. In: Dang TK, Küng J, Chung TM, editors. Proceedings of the Future Data and Security Engineering. Communications in Computer and Information Science; 2025 Nov 27–29; Ho Chi Minh City, Vietnam. Vol. 2708, p. 30–45. doi:10.1007/978-981-95-4721-0_3.
9. Müller M, Jiang Z, Moryossef A, Rios A, Ebling S. Considerations for meaningful sign language machine translation based on glosses. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics; 2023 Jul 9–14; Toronto, ON, Canada. Stroudsburg, PA, USA: Association for Computational Linguistics; 2023. p. 682–93. doi:10.18653/v1/2023.acl-short.60.
10. Núñez-Marcos A, Perez-de-Viñaspre O, Labaka G. A survey on sign language machine translation. Expert Syst Appl. 2023;213(2):118993. doi:10.1016/j.eswa.2022.118993.
11. De Sisto M, Vandeghinste V, Egea Gómez S, De Coster M, Shterionov D, Saggion H. Challenges with sign language datasets for sign language recognition and translation. In: Proceedings of the 13th Language Resources and Evaluation Conference (LREC 2022); 2022 Jun 20–25; Marseille, France. Stroudsburg, PA, USA: European Language Resources Association (ELRA); 2022. p. 2478–87.
12. Amiruzzaman S, Amiruzzaman M, Batchu R, Dracup J, Pham A, Crocker B, et al. Bidirectional translation of ASL and English using machine vision and CNN and transformer networks. Computers. 2026;15(1):20. doi:10.3390/computers15010020.
13. Chen Y, Wei F, Sun X, Wu Z, Lin S. A simple multi-modality transfer learning baseline for sign language translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA. Piscataway, NJ, USA: IEEE; 2022. p. 5120–30. doi:10.1109/CVPR52688.2022.00506.
14. Yin A, Zhong T, Tang L, Jin W, Jin T, Zhao Z. Gloss attention for gloss-free sign language translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 18–22; Vancouver, BC, Canada. Piscataway, NJ, USA: IEEE; 2023. p. 2551–62. doi:10.1109/CVPR52729.2023.00251.
15. Zhang B, Müller M, Sennrich R. SLTUNET: a simple unified model for sign language translation. In: Proceedings of the 11th International Conference on Learning Representations (ICLR 2023); 2023 May 1–5; Kigali, Rwanda. p. 1–18.

16. Zhou B, Chen Z, Clapés A, Wan J, Liang Y, Escalera S, et al. Gloss-free sign language translation: improving from visual-language pretraining. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV); 2023 Oct 2–6; Paris, France. Piscataway, NJ, USA: IEEE; 2023. p. 20871–81. doi:10.1109/ICCV51070.2023.01917.
17. Gong J, Foo LG, He Y, Rahmani H, Liu J. LLMs are good sign language translators. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024 Jun 17–21; Seattle, WA, USA. Piscataway, NJ, USA: IEEE; 2024. p. 18362–72. doi:10.1109/CVPR52733.2024.01739.
18. Wong R, Camgoz NC, Bowden R. Sign2GPT: leveraging large language models for gloss-free sign language translation. In: Proceedings of the 12th International Conference on Learning Representations (ICLR 2024); 2024 May 7–11; Vienna, Austria. p. 1–18.
19. Chen Z, Zhou B, Li J, Wan J, Lei Z, Jiang N, et al. Factorized learning assisted with large language model for gloss-free sign language translation. In: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024); 2024 May 20–25; Torino, Italy. Stroudsburg, PA, USA: ELRA and ICCL; 2024. p. 7071–81. doi:10.18653/v1/2024.lrec-main.620.
20. Hwang EJ, Cho S, Lee J, Park JC. An efficient gloss-free sign language translation using spatial configurations and motion dynamics with LLMs. In: Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies; 2025 Apr 29–May 4; Albuquerque, NM, USA. Stroudsburg, PA, USA: Association for Computational Linguistics; 2025. p. 3901–20. doi:10.18653/v1/2025.naacl-long.197.
21. Jang Y, Raajesh H, Momeni L, Varol G, Zisserman A. Lost in translation, found in context: sign language translation with contextual cues. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2025 Jun 11–15; Nashville, TN, USA. Piscataway, NJ, USA: IEEE; 2025. p. 8742–52.
22. Othman A, Jemni M. English-ASL gloss parallel corpus 2012: ASLG-PC12. In: Proceedings of the LREC2012 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon; 2012 May 27; Istanbul, Turkey. Stroudsburg, PA, USA: European Language Resources Association (ELRA); 2012. p. 151–4.
23. Zhang X, Duh K. Approaching sign language gloss translation as a low-resource machine translation task. In: Proceedings of the 1st International Workshop on Automatic Translation for Signed and Spoken Languages (AT4SSL); 2021 Aug 20; Virtual. Stroudsburg, PA, USA: Association for Machine Translation in the Americas; 2021. p. 60–70. doi:10.18653/v1/2021.mtsummit-at4ssl.7.
24. Chen Y, Zuo R, Wei F, Wu Y, Liu S, Mak B. Two-stream network for sign language recognition and translation. In: Proceedings of the Advances in Neural Information Processing Systems 35 (NeurIPS 2022); 2022 Nov 28–Dec 9; New Orleans, LA, USA. Red Hook, NY, USA: Curran Associates; 2022. p. 17043–56. doi:10.52202/068431-1240.
25. Devlin J, Chang M, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2019 Jun 2–7; Minneapolis, MN, USA. Stroudsburg, PA, USA: Association for Computational Linguistics; 2019. p. 4171–86. doi:10.18653/v1/N19-1423.
26. Kudo T, Richardson J. SentencePiece: a simple and language independent subword tokenizer and detokenizer for neural text processing. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations; 2018 Oct 31–Nov 4; Brussels, Belgium. Stroudsburg, PA, USA: Association for Computational Linguistics; 2018. p. 66–71. doi:10.18653/v1/D18-2012.
27. Koller O, Forster J, Ney H. Continuous sign language recognition: towards large vocabulary statistical recognition systems handling multiple signers. Comput Vis Image Underst. 2015;141(5):108–25. doi:10.1016/j.cviu.2015.09.013.
28. Rampášek L, Galkin M, Dwivedi VP, Luu AT, Wolf G, Beaini D. Recipe for a general, powerful, scalable graph transformer. In: Advances in Neural Information Processing Systems 35 (NeurIPS 2022); 2022 Nov 28–Dec 9; New Orleans, LA, USA. Red Hook, NY, USA: Curran Associates Inc.; 2022. p. 14501–15. doi:10.52202/068431-1054.
29. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015); 2015 May 7–9; San Diego, CA, USA.

30. Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014 Oct 25–29; Doha, Qatar. Stroudsburg, PA, USA: Association for Computational Linguistics; 2014. p. 1724–34. doi:10.3115/v1/D14-1179.
31. Chorowski J, Bahdanau D, Serdyuk D, Cho K, Bengio Y. Attention-based models for speech recognition. In: Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS 2015); 2015 Dec 7–12; Montreal, QC, Canada. Red Hook, NY, USA: Curran Associates Inc.; 2015. p. 577–85.
32. Luong MT, Pham H, Manning CD. Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2015 Sep 17–21; Lisbon, Portugal. Stroudsburg, PA, USA: Association for Computational Linguistics; 2015. p. 1412–21. doi:10.18653/v1/D15-1166.
33. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A, et al. Attention is all you need. In: Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017); 2017 Dec 4–9; Long Beach, CA, USA. Red Hook, NY, USA: Curran Associates Inc.; 2017. p. 5998–6008. doi:10.65215/ctdc8e75.
34. Uzan O, Schmidt CW, Tanner C, Pinter Y. Greed is all you need: an evaluation of tokenizer inference methods. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics; 2024 Aug 11–16; Bangkok, Thailand. Stroudsburg, PA, USA: Association for Computational Linguistics; 2024. p. 813–22. doi:10.18653/v1/2024.acl-short.73.
35. Forster J, Schmidt C, Hoyoux T, Koller O, Zelle U, Piater J, et al. RWTH-PHOENIX-Weather: a large vocabulary sign language recognition and translation corpus. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12); 2012 May 21–27; Istanbul, Turkey. Paris, France: European Language Resources Association (ELRA); 2012. p. 3785–9.
36. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th International Conference on Learning Representations (ICLR 2017); 2017 Apr 24–26; Toulon, France.
37. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. In: Proceedings of the 6th International Conference on Learning Representations (ICLR 2018); 2018 Apr 30–May 3; Vancouver, BC, Canada.
38. Dwivedi VP, Bresson X. A generalization of transformer networks to graphs. arXiv:2012.09699. 2021.
39. Peng N, Poon H, Quirk C, Toutanova K, Yih W. Cross-sentence n-ary relation extraction with graph LSTMs. Trans Assoc Comput Linguist. 2017;5(2):101–15. doi:10.1162/tacl_a_00040.
40. Marcheggiani D, Titov I. Encoding sentences with graph convolutional networks for semantic role labeling. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing; 2017 Sep 7–11; Copenhagen, Denmark. Stroudsburg, PA, USA: Association for Computational Linguistics; 2017. p. 1506–15. doi:10.18653/v1/D17-1159.
41. Bastings J, Titov I, Aziz W, Marcheggiani D, Sima'an K. Graph convolutional encoders for syntax-aware neural machine translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing; 2017 Sep 7–11; Copenhagen, Denmark. Stroudsburg, PA, USA: Association for Computational Linguistics; 2017. p. 1957–67. doi:10.18653/v1/D17-1209.
42. Papineni K, Roukos S, Ward T, Zhu W. BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics; 2002 Jul 7–12; Philadelphia, PA, USA. Stroudsburg, PA, USA: Association for Computational Linguistics; 2002. p. 311–8. doi:10.3115/1073083.1073135.
43. Callison-Burch C, Osborne M, Koehn P. Re-evaluating the role of BLEU in machine translation research. In: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics; 2006 Apr 3–7; Trento, Italy. Stroudsburg, PA, USA: Association for Computational Linguistics; 2006. p. 249–56. doi:10.3115/1608375.1608410.

44. Popović M. chrF: character n-gram F-score for automatic MT evaluation. In: Proceedings of the Workshop on Statistical Machine Translation; 2015 Sep 17–18; Lisbon, Portugal. Stroudsburg, PA, USA: Association for Computational Linguistics; 2015. p. 392–5. doi:10.18653/v1/W15-3049.
45. Post M. A call for clarity in reporting BLEU scores. In: Proceedings of the Third Conference on Machine Translation; 2018 Oct 31–Nov 1; Brussels, Belgium. Stroudsburg, PA, USA: Association for Computational Linguistics; 2018. p. 186–91. doi:10.18653/v1/W18-6319.