

Article

ASL Recognition and Game-Based Interaction: A Machine Learning—Driven, Gamified and Accessible Vocabulary Learning System for Deaf Learners

Stefanie Amiruzzaman ^{1,*}, Raga Mouni Batchu ², Md Amiruzzaman ², Linh Ngo ² and M. Ali Akber Dewan ^{3,*}

¹ Department of Languages and Cultures, West Chester University, West Chester, PA 19383, USA

² Department of Computer Science, West Chester University, West Chester, PA 19383, USA; rbatchu470@wcupa.edu (R.M.B.); mamiruzzaman@wcupa.edu (M.A.); lngo@wcupa.edu (L.N.)

³ School of Computing and Information Systems, Faculty of Science and Technology, Athabasca University, Athabasca, AB T9S 3A3, Canada

* Correspondence: samiruzzaman@wcupa.edu (S.A.); adewan@athabascau.ca (M.A.A.D.)

Abstract

Digital learning tools for American Sign Language (ASL) often lack the interactive depth necessary to engage learners effectively. This paper introduces a novel, browser-based word search game designed to facilitate ASL vocabulary familiarization through gamified interaction. The system employs a two-tier architecture consisting of a React-based frontend and a Flask-based backend. At its core, the application integrates a lightweight, skeleton-based Isolated Sign Language Recognition (ISLR) model, utilizing a Stacked Transformer-based Spatial-Temporal Attention Network to enable real-time webcam-based word entry during the configuration phase. This model, trained on the WLASL-100 dataset, achieves a Top-5 test accuracy of 88.48% with an average model inference latency of 141 ms, enabling real-time webcam input without proprietary hardware. Furthermore, we implement a constraint-satisfaction puzzle generation algorithm that achieves a 100% success rate in creating interlocked, multi-directional grids. Our results demonstrate that merging computer vision with pedagogical game mechanics provides an accessible, high-performance tool for the Deaf and Hard-of-Hearing (DHH) community, bridging the gap between static instruction and active linguistic practice.

Keywords: american sign language; gamified learning; word search puzzle; computer vision; accessibility; educational technology



Academic Editors: Carlos Vaz de Carvalho, Hariklia Tsalapatas and Ricardo Baptista

Received: 17 February 2026

Revised: 26 April 2026

Accepted: 27 April 2026

Published: 7 May 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The proliferation of digital learning platforms has transformed educational access, yet Deaf and Hard-of-Hearing (DHH) learners who rely on American Sign Language (ASL) remain systematically underserved by mainstream educational technology [1,2]. Traditional language-learning applications depend on audio-based instruction and text-centric interfaces, inadvertently creating what researchers term “digital exclusion” for visual language users [3]. This exclusion is particularly problematic given that over 70 million deaf individuals worldwide use sign languages as their primary means of communication, with ASL serving approximately 500,000 to 2 million users in the United States and Canada alone [4]. The mismatch between auditory-centric applications and the visual-gestural nature of ASL represents not merely a usability inconvenience but a fundamental barrier to equitable educational opportunity [2,5,6].

The limitations of conventional computer-assisted language learning (CALL) systems for ASL learners stem from their underlying pedagogical assumptions. Most language-learning platforms are designed for spoken languages, implicitly privileging phonological processing and auditory feedback loops inaccessible to DHH users [7]. Even when such platforms incorporate visual elements, they rarely leverage the spatial-manual modality that constitutes the linguistic foundation of signed languages [8]. Consequently, ASL learners face a paradox: digital tools designed to facilitate language learning often fail to accommodate the linguistic structures that define their target language [5].

This work addresses this critical gap by developing an interactive web application that integrates ASL recognition technology with established pedagogical game mechanics to create an accessible vocabulary learning environment. Our approach is grounded in two complementary theoretical frameworks: embodied cognition, which posits that cognitive processes are rooted in the body's interactions with the world, and sociocultural theory, which emphasizes learning through socially mediated activities [9]. By requiring users to physically produce signs as input, our system engages the same neural and motor pathways used in natural sign language communication, potentially enhancing retention through what cognitive scientists term "enactive encoding" [10].

Our primary contribution is the development of an ASL Word Search Game that transforms passive vocabulary study into active, gamified practice requiring sign production. The system uniquely integrates real-time sign language recognition as its core input mechanism: users "type" words by performing corresponding ASL signs in front of a standard webcam rather than using traditional keyboard or mouse inputs. This gesture-driven interaction paradigm aligns with the visual-gestural nature of ASL, promoting a more natural learning process that respects the linguistic preferences of the DHH community [6,11]. Furthermore, by eliminating the need for specialized hardware, our approach democratizes access to ASL learning technology.

The pedagogical foundation of our system rests on decades of research demonstrating the efficacy of game-based learning for vocabulary acquisition across multiple languages and learner populations [12–14]. Game mechanics such as goal-setting, immediate feedback, and progressive challenge have been shown to enhance learner motivation and improve long-term retention compared to drill-based methods [15]. Word search puzzles, in particular, offer specific cognitive benefits: they require sustained visual attention, pattern recognition, and orthographic processing [13,14]. Studies with hearing learners have demonstrated that word search activities improve spelling accuracy and word recognition through repeated, contextually rich exposure [16]. Our system extends these benefits to ASL learning by replacing text-based search with sign-based interaction, creating what we term "gesture-driven gamification", a synthesis of embodied interaction and educational game design.

Recent technological advances have made this synthesis possible. The field of sign language recognition has evolved from laboratory-bound systems requiring specialized sensors to lightweight, real-time models deployable in standard web browsers [5,17]. The development of the WLASL dataset [18] provided researchers with a standardized benchmark for isolated sign recognition, catalyzing improvements in model architecture and training methodology. Subsequent innovations, including pose-based graph convolutional networks [19], transformer architectures adapted for sign language [20], and spatial-temporal attention mechanisms [21], have progressively pushed the boundaries of recognition accuracy while reducing computational requirements. Our system builds directly upon these advances, incorporating a Stacked Transformer-based Spatial-Temporal Attention Network [22] optimized for real-time browser deployment.

The bidirectional translation system proposed by Amiruzzaman et al. [21] demonstrated the feasibility of ASL-English conversion using machine vision and CNN-Transformer networks, achieving significant improvements in recognition accuracy while maintaining practical inference speeds. Our work extends this technological foundation into the pedagogical domain, adapting state-of-the-art recognition capabilities for educational purposes. This adaptation requires careful consideration of the distinct demands of learning applications: tolerance for recognition ambiguity (through top-5 predictions), integration with game mechanics, and responsiveness sufficient for fluid interaction.

The technical implementation of our system leverages recent breakthroughs in lightweight machine learning to deliver a complete browser-based solution requiring no proprietary hardware or specialized installation. Our Isolated Sign Language Recognition (ISLR) model, based on a Stacked Transformer architecture trained on the WLASL-100 dataset [18], achieves competitive accuracy (88.48% Top-5) while maintaining an average model inference latency of 140.45 ms, well within the threshold for real-time interactive applications [23]. The puzzle generation subsystem employs constraint satisfaction techniques adapted from crossword construction algorithms [24,25] to create challenging, interconnected grids that maximize learning opportunities through strategic word placement and inter-word intersections. The algorithm achieves 100% success rate across diverse input configurations while maintaining generation times under 0.2 s even for large grids.

The primary scientific novelty of this work lies in the architectural and pedagogical synthesis of a classic game with advanced computer vision. While word search puzzles are proven vocabulary-building tools, adapting them specifically for ASL learning is a uniquely effective approach that respects the visual-gestural nature of the language. We elevate this unique pedagogical tool by integrating it within a custom, two-tier architecture powered by a lightweight Stacked Transformer Spatial-Temporal Attention Network. By executing real-time skeletal recognition with a 139 ms inference latency directly in the browser, the system allows students to actively ‘type’ puzzle answers via physical sign production without specialized hardware. Combining this tailored puzzle format with low-latency, gesture-driven gamification transforms ASL vocabulary acquisition from passive observation into an accessible, highly interactive linguistic practice.

This paper makes three principal contributions to the fields of accessible educational technology and sign language recognition:

1. **Pedagogical Innovation:** The design and implementation of a fully accessible web application that uses ASL signing as the primary input mechanism for a word search game, establishing a new paradigm for gesture-driven language learning that respects the visual-gestural nature of signed languages.
2. **Technical Advancement:** A lightweight ISLR model achieving 88.48% Top-5 accuracy with 140.45 ms model inference latency, specifically optimized for real-time browser deployment through skeleton-based processing and model compression techniques, making advanced sign recognition accessible without specialized hardware.
3. **Algorithmic Contribution:** A novel constraint-based puzzle generation algorithm achieving a very good success rate across diverse input configurations, incorporating diagonal placement capabilities and intersection-first placement strategies that produce visually rich, educationally effective puzzles while maintaining real-time performance.

Beyond these specific contributions, our work demonstrates the broader feasibility of integrating computer vision technologies with educational game design to create inclusive learning tools that serve historically marginalized populations. The positive results from our user study with 25 participants, including statistically significant gains in learner confi-

dence ($p < 0.001$), suggest that gesture-driven gamification merits further investigation as a pedagogical approach for signed language instruction.

The remainder of this paper is organized as follows: Section 2 provides a review of related work spanning gamified language learning, accessible technology for DHH users, sign language recognition systems, and the machine learning foundations of modern ISLR. Section 3 details our system architecture, including the client-server implementation, webcam-based recognition pipeline, and puzzle generation algorithms. Section 4 presents a rigorous evaluation of both the recognition model's performance, the puzzle users' reliability, and the findings from our user study, including analyses of usability, engagement, and learning outcomes. Finally, Section 5 offers conclusions, acknowledges limitations, and outlines directions for future research, including vocabulary expansion, adaptive difficulty mechanisms, and integration with formal ASL curricula.

2. Related Work

The foundation of this project is built upon three complementary research areas: the use of gamification in language learning, the development of accessible educational tools for Deaf learners, and the integration of machine learning for sign language recognition. We review prior contributions across these domains to establish the context and rationale for our proposed system.

2.1. Gamification and Vocabulary Acquisition

The efficacy of *game-based learning* in language education, particularly for vocabulary mastery, is a prominent area of pedagogical research. Studies have consistently demonstrated that integrating interactive, competitive, or puzzle-based activities can significantly enhance learner engagement, motivation, and retention compared to conventional methods [13,14]. Word search puzzles, specifically, are shown to be effective tools for reinforcing spelling patterns, promoting repeated word exposure, and fostering a more enjoyable atmosphere, which directly contributes to measurable improvements in vocabulary test scores [13,14]. The benefits reported in this research, such as reduced boredom and increased active participation, justify the selection of the word search format for this project as a mechanism to facilitate active ASL vocabulary familiarization.

To operationalize this pedagogical approach, our system features a custom puzzle generator designed to create these word searches on the fly, allowing learners to continuously practice the ASL alphabet and target vocabulary. While foundational word search and crossword construction algorithms exist, such as the WoSeCon algorithm described by Moussiades [26] and the unconstrained crossword automation strategies proposed by Agarwal and Joshi [27], we adapt and extend these concepts for real-time educational use. Specifically, our generator builds upon these established spatial allocation strategies while explicitly incorporating constraint-based diagonal placements, a structural feature that many standard pedagogical generators do not support.

2.2. Accessible Learning for Deaf and Hard-of-Hearing Users

A crucial motivation for this project stems from the need for educational technology that is intrinsically accessible to the *Deaf and Hard-of-Hearing (DHH) community*. Many standard educational games rely on auditory components or written language alone, which can be restrictive for ASL users whose primary language is visual-gestural [28,29]. Early work by Brashear et al. (2006) [11] provided influential evidence that *sign-based interfaces* could support educational gameplay by allowing children to control game actions directly through ASL signs, demonstrating that gesture-driven interactions lead to a more natural and engaging learning experience. This concept was reinforced by Chebka and

Essalmi (2015) [28], who designed a crossword-style game that strategically leveraged visual communication and integrated sign-language videos and visual clues to support vocabulary acquisition, confirming that learning environments aligned with DHH learners' linguistic and sensory strengths enhance literacy development.

The pedagogical design of this system, which transitions from sign-based word input to alphabet-based search, is strategically aligned with natural language acquisition patterns. Research in Deaf education, most notably by Locke [30] and Meier and Newport [31], demonstrates that Deaf children typically acquire high-frequency conceptual signs as early as 6 to 9 months, significantly before mastering the 26 distinct handshapes of the manual alphabet. This lexical priority is also observed in hearing populations, where children develop an extensive spoken vocabulary before formal orthographic training. By utilizing a webcam-based Isolated Sign Language Recognition (ISLR) model for word entry, the tool allows learners to leverage existing sign knowledge to practice the more abstract manual alphabet in a gamified context. This approach is further supported by Padden [32], who argues that early learners often perceive fingerspelling as holistic lexical units rather than individual letters. Consequently, the game serves as a bridge for both DHH and hearing users who may know ASL words but have not yet mastered the alphabet, transforming fingerspelling from a rote-memorization task into a functional, interactive component of language learning. These findings underscore the importance of designing systems that replace auditory or text-centric input with a visual-gestural mechanism, such as the ASL fingerspelling recognition proposed here.

The prototype presented in this paper can be readily modified to manage access levels for both students and instructors. The primary objective of this work is to showcase the underlying concept and demonstrate that such a system serves as an effective tool for students to practice the ASL alphabet in an engaging and accessible manner.

2.3. ASL Recognition Technology in Interactive Systems

The application of computer vision and machine learning to sign language has rapidly advanced, moving from early, hardware-intensive systems to lightweight, accessible platforms. Foundational work on integrating sign-language recognition into games, such as that by Brashear et al. (2006) [11], used techniques like Hidden Markov Models (HMMs) to recognize gestures in real time, though these systems were often limited in vocabulary and recognition accuracy. More recently, researchers have focused on deploying *efficient, browser-based recognition* and leveraging gamification for both practice and data collection. Bragg et al. (2021) [33] developed "ASL Sea Battle", a game designed to collect large, high-quality ASL datasets, demonstrating that the inclusion of gameplay mechanics (like goals and competition) is vital for sustained user engagement and data quality. Building on this trend toward lightweight deployment, Medronha et al. (2024) [17] introduced LERMO, a browser-based game inspired by Wordle (a popular daily web-based word-guessing puzzle), that successfully integrates real-time fingerspelling recognition via webcam input. Their work not only proves the feasibility of deploying efficient sign-language classifiers directly in a standard web environment but also confirms the high usability and strong perceived educational value of such *AI-driven, gamified tools* for fingerspelling practice. Our project draws directly from the successes of LERMO in utilizing accessible, web-based recognition to create an engaging learning application.

Technical approaches to sign language recognition have diversified significantly. Pose-based methods using Graph Convolutional Networks (GCNs) have shown particular promise for their efficiency and robustness [18]. Transformer architectures have recently been adapted for sign language recognition, with models like SignBERT [34] achieving state-of-the-art performance through self-supervised pre-training. As summarized in Table 1,

our system builds upon this progression by implementing a word-search interface powered by real-time Isolated Sign Language Recognition (ISLR).

Table 1. Evolution of Sign Language Recognition in Educational Applications.

System	Key Contribution	Year
ASL Sea Battle [33]	Game-based ASL data collection	2021
LERMO [17]	Browser-based fingerspelling game	2024
SignBERT [34]	Self-supervised sign representation	2023
SSTAN [35]	Spatial-temporal attention network	2025
Our System	Word search with real-time ISLR	2026

2.4. AI/ML in ASL Recognition

The domain of American Sign Language (ASL) recognition via AI/ML is primarily stratified into two core tasks based on the complexity of temporal boundaries: Isolated Sign Language Recognition (ISLR) and Continuous Sign Language Recognition (CSLR). ISLR functions similarly to standard video classification, where the objective is to assign a single class label (a gloss, representing a complete conceptual word rather than a single fingerspelled letter) to a pre-trimmed video clip containing exactly one sign. Recent approaches typically utilize 3D Convolutional Neural Networks (3D-CNNs) or Graph Convolutional Networks (GCNs) to extract spatial-temporal features from these short clips.

In contrast, most practical settings require Continuous Sign Language Recognition (CSLR), where the input is an untrimmed signing stream (a continuous video showing a sequence of multiple signs performed fluidly without explicit pauses) and the output is a gloss sequence. The central difficulty is temporal alignment, deciding when each sign occurs without reliable boundary annotations, so CSLR systems typically learn sequence mappings that can tolerate variable-length segments and ambiguous transitions. A crucial cross-cutting capability is Non-Manual Marker (NMM) Detection, which focuses on facial expressions, head pose, and mouthing, since these cues can encode grammar (e.g., question marking) and modulate meaning. Accordingly, many modern pipelines treat non-manual signals as first-class inputs and combine them with manual hand/arm information in a multi-stream or multi-modal representation [36].

Moving beyond recognition, the field extends to translation and generation. Sign Language Translation (SLT) aims to produce fluent spoken-language sentences from signing input, emphasizing grammatical correctness and meaning preservation rather than a word-by-word gloss listing. Conversely, Sign Language Production (SLP) tackles the inverse direction: generating sign from text. SLP approaches often predict intermediate motion representations (e.g., keypoints or avatar control signals) and then render them as video or animation [37].

Dataset choices depend mainly on clip granularity and annotation type: ISLR datasets provide trimmed sign clips with class labels, CSLR datasets provide longer sequences with gloss annotations, SLT datasets add paired spoken-language translations, NMM-focused datasets emphasize facial/head/mouth cues, and SLP work often relies on pose/keypoint representations for motion generation. Table 2 summarizes commonly used datasets for each task.

Table 2. Common datasets used across sign language AI/ML tasks.

Task	Representative Datasets
Isolated Sign Language Recognition (ISLR)	WLASL [18]; MS-ASL [38]; ASL Citizen [39]
Continuous Sign Language Recognition (CSLR)	RWTH-PHOENIX-Weather 2014T [40]; CSL-Daily [41]; Bi-directional ASL [21]
Non-Manual Marker Detection (NMM)	NMFs-CSL [42]; How2Sign [43]
Sign Language Translation (SLT)	RWTH-PHOENIX-Weather 2014T [40]; CSL-Daily [41]; How2Sign [43]
Sign Language Production (SLP)	PHOENIX-2014T [40] (pose-derived); How2Sign (3D pose subsets) [43]

Sign language recognition models are often categorized by input modality, ranging from RGB-based approaches that operate directly on pixels and can capture rich visual cues, to skeleton-based methods that represent the signer as compact landmark graphs and can be more efficient and background-invariant. Many modern systems combine multiple streams, but the practicality of skeleton-based pipelines depends heavily on the pose/landmark extractor. OpenPose is a well-known bottom-up estimator: it predicts part confidence maps and associates parts to individuals using Part Affinity Fields (PAFs), which is effective in multi-person scenes but can be computationally demanding [44]. In contrast, MediaPipe Holistic is designed for real-time, largely single-subject use cases and follows a detect–track style pipeline: a detector initializes (or re-initializes) a region of interest, after which lightweight landmark models run per frame to maintain stable tracking of the full set of holistic landmarks [45].

2.5. Isolated Sign Language Recognition

In Isolated Sign Language Recognition (ISLR), early deep-learning systems largely followed a “strong visual backbone + simple temporal aggregation” recipe. As larger benchmarks appeared, researchers gravitated toward spatiotemporal CNNs that could model motion directly from RGB clips. A canonical milestone for ASL ISLR is the WLASL dataset paper by Li et al., which helped standardize large-vocabulary word-level evaluation and popularized strong video baselines such as 3D for word-level ASL recognition [18]. In this stage, most performance gains came from better video backbones (2D → 3D CNNs), better crops/augmentations, and more data, while temporal reasoning was often implicit in the backbone or handled with lightweight sequence heads.

As pose/keypoint extraction improved, a second line of work pushed ISLR toward skeleton-based representations that encode signing as joint trajectories rather than pixels. This family benefited from the broader action-recognition community’s move to graph-based skeleton modeling, especially ST-GCN proposed by Yan et al. (2018), which introduced spatiotemporal graph convolutions over human joints and inspired many “pose as a graph” formulations [46]. A representative evolution within ISLR then replaced fixed graph connectivity with attention mechanisms: instead of assuming a hand-crafted adjacency, Transformers let the model learn which joints/frames matter for a given sign instance. For example, Boháček and Hrůz proposed a pose-based Transformer for word-level sign recognition, reflecting the broader shift from GCN/RNN-style temporal modeling to attention-first sequence modeling for keypoints [47].

Transformer designs have been refined to better reflect the structure of signing by explicitly separating intra-frame (spatial) from inter-frame (temporal) reasoning. Hirooka et al. (2025) proposes a sequential spatial–temporal attention architecture aimed at capturing

joint relations within each frame and long-range dynamics across frames, and reports strong performance on fingerspelling datasets and competitive results on WLASL [35].

More recently, the field has leaned heavily into pretraining to combat limited labeled sign data and improve generalization across signers/environments. A prominent example is SignBERT, which pretrains on hand/pose-derived “visual tokens” using self-supervised masking objectives and then fine-tunes on downstream SLR/ISLR benchmarks (including WLASL/MSASL), demonstrating that language-model-like pretraining ideas transfer well to sign representations [34].

3. Method

This section describes the technical architecture and implementation of the ASL Word Search Game, including the system design, puzzle-generation algorithms, word-validation mechanisms, and the integration of real-time sign-language recognition. The goal of the system is to provide an accessible, browser-based platform where users can practice ASL vocabulary through an interactive word-search puzzle powered by lightweight machine-learning components.

3.1. System Architecture Overview

The “ASL Word Search” game is implemented via a modular, two-tier architecture comprising a React-based client application and a lightweight Flask backend service. Figure 1 illustrates the high-level system design, while the initial user interface in default mode is depicted in Figure 2. The system functionality is partitioned into two primary layers: the frontend and the backend. The frontend manages all user-facing interactions, including rendering the puzzle grid, capturing webcam video streams, and processing configuration inputs (see Figure 3). Concurrently, the backend is responsible for the core computational logic, specifically the generation of word-search puzzles and the execution of real-time Sign Language recognition.

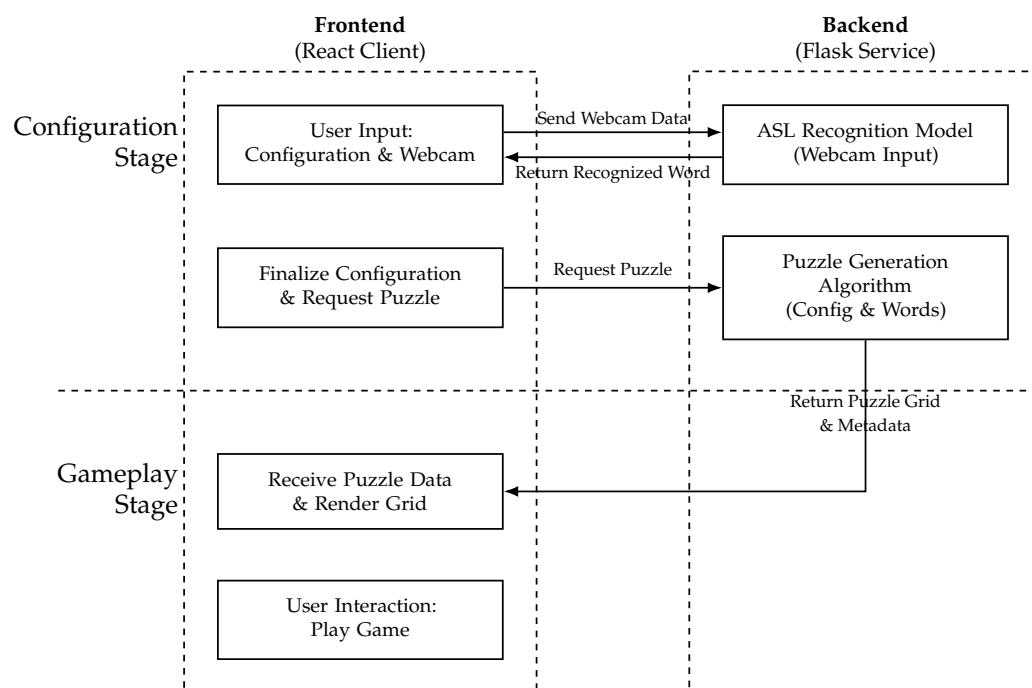


Figure 1. High-level system architecture of the ASL Word Search Game, illustrating the interaction between the React frontend, Flask backend, ISLR recognition module, and puzzle generation engine.

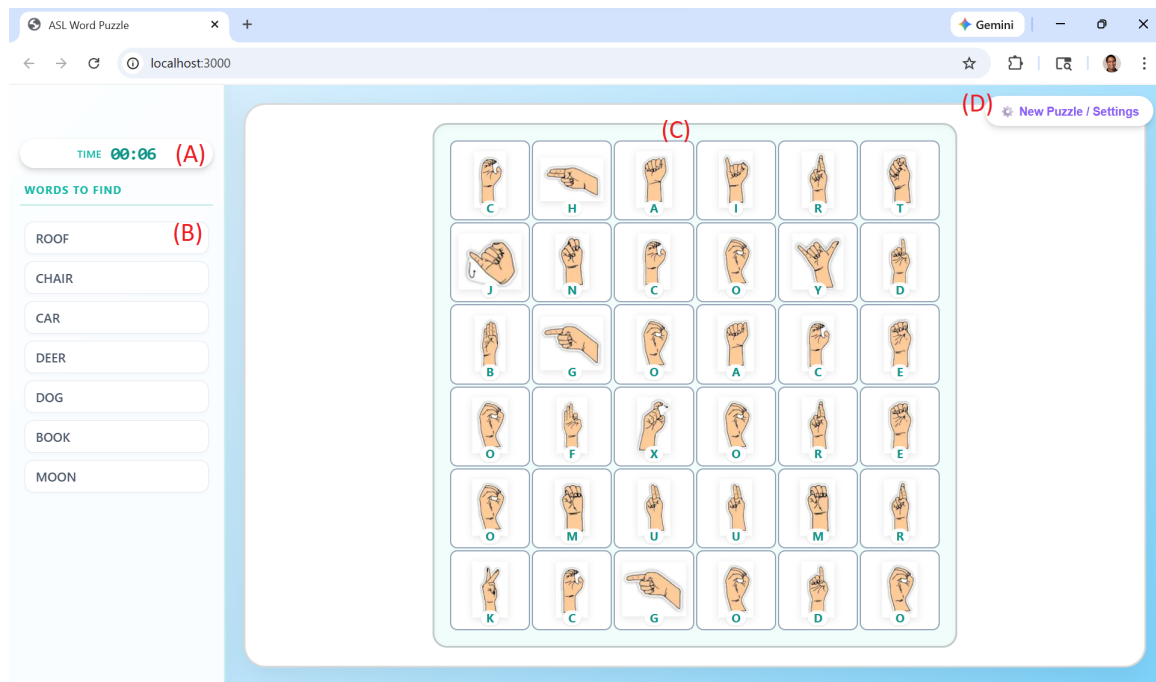


Figure 2. Main gameplay interface of the ASL Word Search Game, displaying (A) showing the time they take to complete the game. Once a user completes the game, then they will see how long it took them to complete the game. (B) the word list they need to find, (C) showing the word puzzle that they need to solve, (D) settings that allow users to start a new game or customize a game.

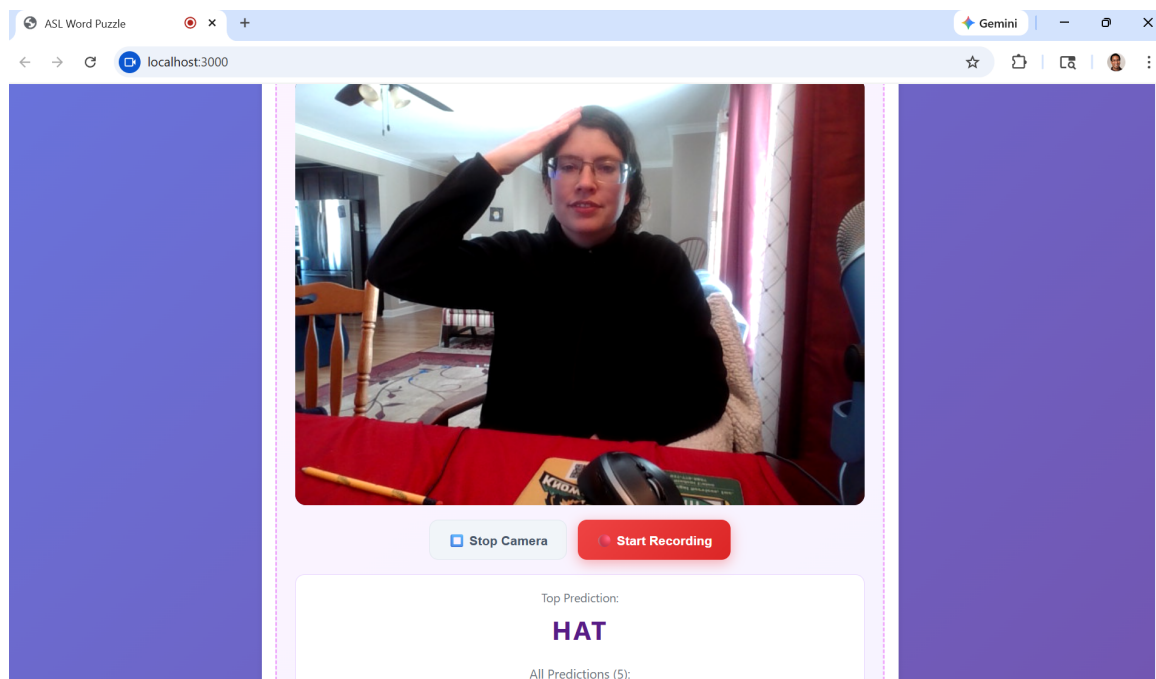


Figure 3. Option for users to enter words by using ASL signs via the webcam. In this example, a user is signing “hat” the system recognizes and shows the word as a possible choice.

Furthermore, the frontend operates as a single-page React application that manages the overall game flow, coordinating transitions between the configuration view (SetupPage) and the interactive puzzle gameplay view (GamePage). In the configuration stage, users specify puzzle parameters such as grid size, number of words, and custom vocabulary (see Figure 4). Once these selections are submitted, the frontend requests a new puzzle from the backend and shifts the interface from setup to gameplay. During the gameplay phase, the

frontend maintains the puzzle state locally, renders the grid dynamically, and tracks the user's word selections to provide immediate feedback and interaction.

Figure 4. The settings window, which allows both manual word entry and ASL sign-based user input for the word choices. Users have a choice of going with a simple or a custom option. The simple option goes with the default grid and word choices, whereas the custom option allows the user to select the grid size and enter the words of their choice.

For custom puzzles, users have multiple input methods:

1. **Manual text entry:** Instructors can type or paste word lists into a text field. Words may be entered comma-separated or one per line; the system parses and normalizes them (e.g., trimming and uppercasing) and shows a preview of the parsed list before puzzle generation. A user can also use ASL alphabet signs (see Figure 5).
2. **File upload:** A text file (e.g., .txt or .csv) can be uploaded, with one word per line or comma-separated Figure 6. This supports reusing lists (e.g., vocabulary sets or curriculum word lists) and optionally limiting how many words are drawn from the file for a single puzzle.
3. **ASL sign builder:** Users build words by selecting letters from an on-screen set of ASL alphabet signs (see Figure 5). Clicking a sign appends that letter to the current

word; the word can then be added to the puzzle word list. This supports vocabulary selection while reinforcing letter–sign associations.

4. Webcam-based ASL recognition: Users can record themselves signing a word via the browser’s camera Figure 3. The video is sent to a backend ASL recognition model; the top predicted word is shown and can be added to the custom word list. If recognition fails or is unavailable, manual text entry is offered as a fallback. This allows instructors or learners to define puzzle words by signing rather than typing.

Puzzle Configuration

Grid Size:

Grid size must be between 5 and 8

Number of Words to Include (optional):

Randomly selects this many words from your list

☐ Enter words manually
 ☐ Upload word file
 ☒ Use ASL sign builder
 ☐ Record ASL via Webcam

ASL Sign Builder Current word: —

Click a sign to add its letter. Build a word, then add it to your list.

 A	 B	 C	 D	 E	 F	 G
 H	 I	 J	 K	 L	 M	 N
 O	 P	 Q	 R	 S	 T	 U

Select Learning Mode

Beginner
Shows English letters below ASL signs

Advanced
Only ASL signs (no hints)

START GAME

Figure 5. ASL alphabet-based word builder allowing users to construct vocabulary through visual letter selection.

Together, these options support different workflows (quick typing, bulk file upload, sign-based building, and sign recognition) and make the tool usable with or without a working ASL recognition backend.

The backend is implemented as a lightweight Flask service that supports two primary computational components: puzzle generation and real-time sign-language recognition. In

the configuration stage, the backend can accept webcam-derived input from the frontend and use it to help users build a custom word list.

As illustrated in Figure 7, this ASL pipeline takes raw webcam video, extracts skeletal hand and upper-body landmarks using the MediaPipe Python package, applies pre-processing to form short temporal sequences, and feeds these sequences into a temporal transformer model trained on the WLASL-100 dataset [18]. The model is first trained offline on WLASL-100, and the resulting learned parameters are stored in a .pth weights file. During real-time inference, these saved weights are loaded to run the transformer model efficiently, allowing it to predict the signed word from the incoming video stream. The predicted word is then returned to the frontend so that users can add it directly to their custom vocabulary.

Figure 6. File upload option: CSV-based vocabulary upload feature enabling instructors or learners to define custom word lists.

Regarding display requirements, the application is designed to be fully responsive across both desktop and mobile devices. Leveraging React’s component-based architecture enables the frontend to dynamically scale the puzzle grid and interface to accommodate various screen sizes without sacrificing functionality. On smartphones, the system operates seamlessly via standard mobile web browsers, such as Google Chrome and Apple Safari. While larger displays (e.g., desktop monitors or tablets) offer an optimal viewing experience for larger grids and allow users to easily maintain the physical distance required for accurate upper-body

capture during webcam-based word entry, this responsive mobile implementation ensures the core puzzle game remains highly accessible across everyday devices.

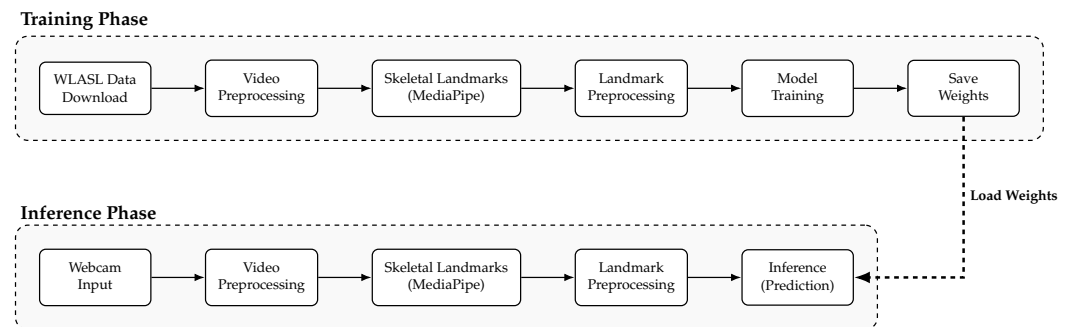


Figure 7. End-to-end ASL word prediction pipeline showing both offline model training on WLASL-100 and real-time inference from webcam input (arrows are showing the flow directions).

As shown in Figure 8, once the configuration is finalized (grid length, number of words, and custom words), the backend invokes a heuristic, constraint-based word search generation algorithm that uses directional role assignment (e.g., straight vs. diagonal words), intersection-first placement to maximize overlaps, and randomized fallback strategies when constraints cannot be satisfied. The result is a completed puzzle grid and associated metadata, which are sent back to the client to initialize gameplay.

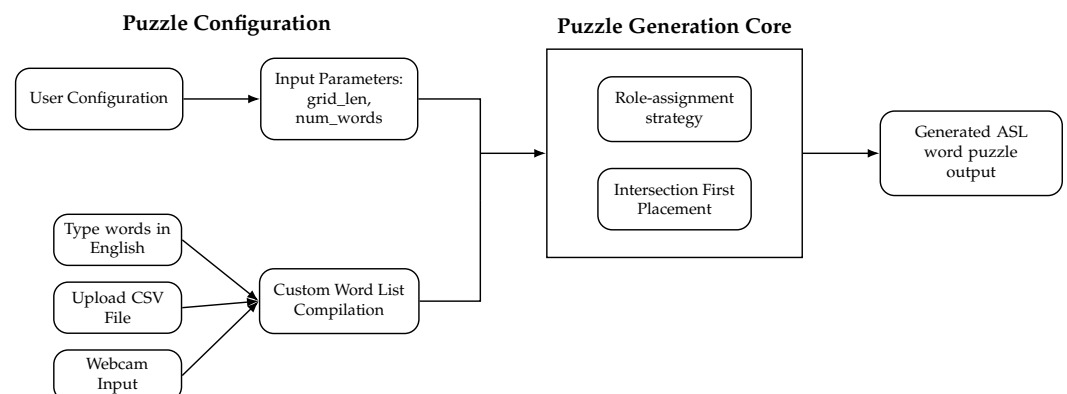


Figure 8. Constraint-based puzzle generation workflow illustrating user configuration inputs, role assignment, intersection-first placement, and final grid construction.

3.2. Webcam-Based Word Entry via Isolated Sign Language Recognition

As described in Section 3.1, our puzzle game supports multiple modalities for word entry. One such modality enables users to input words through webcam-based sign language recognition. Since the puzzle operates exclusively at the word level, the recognition task can be naturally formulated as Isolated Sign Language Recognition (ISLR) rather than continuous sign or fingerspelling recognition. This formulation allows each captured sign clip to be mapped directly to a single lexical unit, which aligns well with the interaction design of the game.

As discussed in Section 2.4, the WLASL dataset is well-suited for this task, as it provides a large vocabulary of isolated American Sign Language (ASL) signs with corresponding video samples. However, WLASL contains over 2000 unique word classes, and training a highly accurate classifier over the full label space is challenging, particularly in real-time and interactive settings [18]. Prior work similarly constrains the vocabulary to smaller subsets to achieve reliable performance. Following this practice, we limit the recognition space to a curated subset of 100 most commonly used words. To further improve

usability and robustness, the system presents the top-5 predicted words instead of a single prediction, allowing users to select the intended word when ambiguity arises.

We adopt the Stacked Transformer-based Spatial–Temporal Attention Network as the backbone of our recognition model, as it represents one of the strongest architectures for skeleton-based isolated sign recognition [35]. The model operates on pose sequences extracted from video, enabling it to capture both spatial relationships between joints and their temporal dynamics across frames. We used the model as a foundation and adapted it to support real-time inference from webcam input. This includes structuring the model to operate on short, fixed-length pose sequences and ensuring that inference latency remains low enough for interactive use. By relying on skeletal representations rather than raw RGB frames, the model remains computationally efficient and robust to background variation, lighting changes, and signer appearance.

To support real-time webcam input with low latency, we extract skeleton landmarks using MediaPipe Holistic Figure 9 (pose + both hands) rather than OpenPose, which reduces runtime overhead and allows the pipeline to run in CPU-only settings. For each frame, we collect 33 pose and 21×2 hand landmarks, along with visibility/presence masks; frames without reliable detections are encoded as zeros to avoid introducing noise. We then normalize the skeleton by translating joints relative to a body reference point (mid-hip) and scaling by shoulder width for basic view/size invariance, and store sequences as .npz files (skeleton + mask).

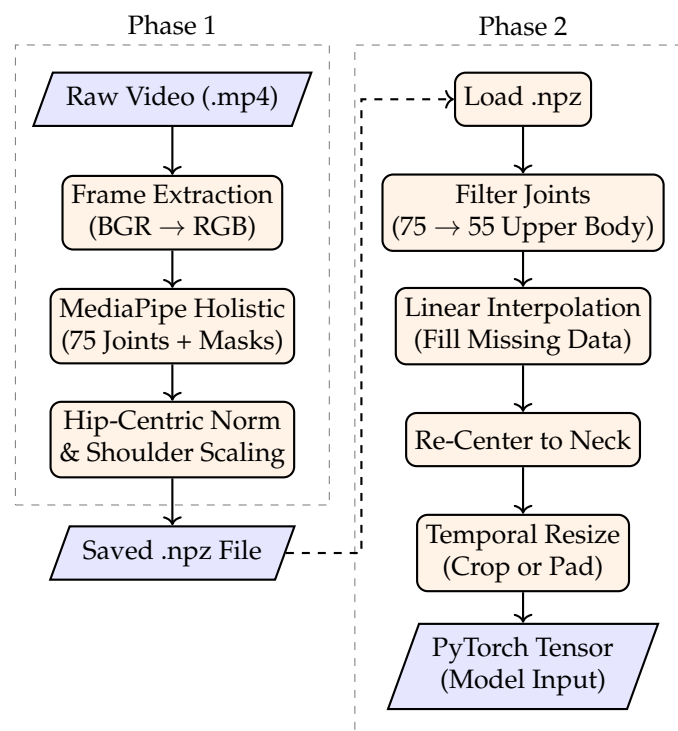


Figure 9. Two-phase video preprocessing pipeline for skeleton-based model training, including landmark extraction, normalization, filtering, and temporal resizing.

During dataset loading, we keep all hand landmarks while using a compact subset of upper-body pose joints (reducing the representation from 75 to 55 joints), interpolate missing values over time, center around the neck (mid-shoulder), and enforce a fixed clip length (e.g., 50 frames) via random/center cropping and padding by repeating the last frame. Light geometric augmentation (small shifts/rotations) is applied during training to improve robustness under natural webcam motion.

Our recognition backbone is adapted from a stacked spatial-temporal attention design, comprising 10 sequential spatial-temporal attention (SSTAN) blocks. Skeleton joint coordinates are projected into an embedding space of $D_{model} = 128$, processed via 8-head spatial and temporal attention mechanisms, and passed through a temporal feed-forward network of $D_{ffn} = 512$ (with a 0.25 dropout rate). To capture structural and sequential order, we employ a hybrid positional encoding strategy. For the intra-frame spatial dimension, fixed sinusoidal encodings and a unified learnable class token are added to the projected embeddings. For the inter-frame temporal dimension, a learnable 1D relative position bias table is integrated directly into the attention scores, enabling the network to generalize robustly across varying sequence lengths by focusing on relative frame-to-frame distances. For classification, the CLS representation is aggregated using mask-aware temporal pooling to ignore padded frames, and a final linear head predicts the target word class. Training uses AdamW, cross-entropy with label smoothing, and a cosine learning-rate schedule with warmup.

At inference time, pose and hand landmarks are streamed from the webcam and accumulated into short, fixed-length skeleton clips using the same preprocessing pipeline as training. Each clip is passed through the trained model in a sliding-window manner, enabling near real-time prediction while maintaining temporal consistency. Instead of outputting a single class label, the model returns the top-5 word predictions ranked by confidence, allowing users to select the intended word when ambiguity occurs.

Our system intrinsically mitigates environmental variability such as lighting, background complexity, and resolution through its skeleton-based architecture. Because the spatial-temporal transformer operates exclusively on normalized skeletal landmarks rather than raw RGB pixels, downstream classification is invariant to visual noise. We delegate environmental robustness to the MediaPipe Holistic feature extractor, which is heavily optimized for diverse real-world webcam conditions. Provided the lighting allows MediaPipe to initialize the pose graph, our model maintains consistent predictive accuracy.

3.3. Puzzle Generation

The ASL Word Search generation process follows a four-stage pipeline designed to promote interlocked word placement while ensuring bounded execution.

In the planning stage, ASL-PREPARE (1) ranks input words according to their intersection potential (see Algorithm 1). Specifically, for each word, a ranking score is computed as the sum of the number of distinct letters shared with every other word in the set. Words are then sorted in descending order of this rank, with longer words prioritized in case of ties. To introduce structural constraints, a randomly selected fraction (approximately 20–50%) of words are assigned diagonal placement roles, while the remaining words are restricted to orthogonal orientations (see Algorithm 1).

The ordered list is passed to ASL-CONSTRUCT (2), which governs the primary placement loop (see Algorithm 2). Words are processed sequentially, and for each word a placement attempt is made. If placement fails, the algorithm performs bounded backtracking by removing the most recently placed word and retrying alternative configurations. This process is limited by a maximum backtracking budget B , ensuring termination and preventing combinatorial explosion (see Algorithm 2).

For each placement attempt, ASL-PLACE (3) is invoked. This subroutine first performs an intersection-first search, attempting to align the new word with matching letters already present in the grid (see Algorithm 3). If no valid intersection-based placement is found, the algorithm falls back to randomized placement attempts constrained by the word's assigned orientation (diagonal or orthogonal).

Finally, ASL-FINALIZE (4) completes the puzzle grid by replacing all remaining empty cells with randomly generated letters, producing the final word search configuration (see Algorithm 4).

Algorithm 1 ASL-PREPARE: Word Ranking and Role Assignment

```

1: Input:  $S$  (list of input words),  $N$  (grid size)
2: Output:  $W$  (sorted list of Word States),  $G$  (Initialized Grid)
3: function ASLPREPARE( $S, N$ )
4:    $S' \leftarrow$  cleaned version of  $S$  (uppercase, trimmed, length  $\geq 2$ , deduplicated)
5:   if  $S' = \emptyset$  then
6:     return  $\emptyset$ , empty  $N \times N$  grid  $G$ 
7:   end if
8:   Initialize  $G$  as  $N \times N$  filled with - ▷ Calculate Ranking  $R$ 
9:   for all  $w \in S'$  do
10:     $R[w] \leftarrow 0$ 
11:  end for
12:  for all pairs  $(w_i, w_j), i < j$  in  $S'$  do
13:     $R[w_i] \leftarrow R[w_i] + |\text{letters}(w_i) \cap \text{letters}(w_j)|$ 
14:     $R[w_j] \leftarrow R[w_j] + |\text{letters}(w_i) \cap \text{letters}(w_j)|$ 
15:  end for ▷ Sort  $L$  by rank and length
16:   $L \leftarrow S'$  sorted by  $(R[w], |w|)$  in descending order ▷ Assign Roles
    (Diagonal/Orthogonal)
17:  Sample diagonal ratio in  $[0.2, 0.5]$  and choose that fraction of indices as DiagIndices
18:   $W \leftarrow \emptyset$ 
19:  for  $k \leftarrow 0$  to  $|L| - 1$  do
20:     $W[k] \leftarrow$  (word =  $L[k]$ , is_diag =  $(k \in \text{DiagIndices})$ , tried =  $\emptyset$ )
21:  end for
22:  return  $W, G$ 
23: end function

```

Algorithm 2 ASL-CONSTRUCT: Placement and Limited Backtracking

```

1: Input:  $W$  (sorted Word States),  $G$  (initial grid),  $N$  (grid size),  $B$  (backtracking budget)
2: Output: Partially completed grid  $G$ 
3: function ASLCONSTRUCT( $W, G, N, B$ )
4:   Initialize Occupancy as  $N \times N$  of zeros
5:    $Stack \leftarrow \emptyset, b \leftarrow 0, i \leftarrow 0$  ▷ Main construction loop
6:   while  $0 \leq i < |W|$  and  $b < B$  do
7:      $ws \leftarrow W[i]$ 
8:      $placed \leftarrow \text{TRYPLACE}(ws, G, \text{Occupancy}, \text{Stack})$ 
9:     if  $placed$  then
10:      if  $i = |W| - 1$  then
11:        break
12:      end if
13:       $i \leftarrow i + 1$  ▷ Move forward
14:    else
15:      if  $i = 0$  then
16:        break ▷ Cannot backtrack further
17:      end if
18:       $W[i].\text{tried} \leftarrow \emptyset$ 
19:       $(w, s, d) \leftarrow \text{POP}(\text{Stack})$ 
20:       $\text{REMOVWORD}(w, s, d, G, \text{Occupancy})$ 
21:       $b \leftarrow b + 1; i \leftarrow i - 1$  ▷ Backtrack one word
22:    end if
23:  end while
24:  return  $G$ 
25: end function

```

Algorithm 3 ASL-PLACE: Word Placement Attempt Strategy

```

1: Input: ws (current Word State), G (grid), Occupancy, Stack
2: Output: true if placed, false otherwise
3: function TRYPLACE(ws, G, Occupancy, Stack)
4:   word  $\leftarrow$  ws.word, T  $\leftarrow$  ws.tries
5:   if ws.is_diag then
6:     D  $\leftarrow$   $\{(1, 1), (1, -1), (-1, 1), (-1, -1)\}$ 
7:   else
8:     D  $\leftarrow$   $\{(0, 1), (0, -1), (1, 0), (-1, 0)\}$ 
9:   end if
10:  Randomly permute D ▷ Intersection-first search
11:  for each letter index idx of word in random order do
12:    for each cell (r, c) with G[r, c] = word[idx] in random order do
13:      for all (dr, dc)  $\in$  D do
14:        s  $\leftarrow$  (r − idx · dr, c − idx · dc); key  $\leftarrow$  (s, dr, dc)
15:        if key  $\notin$  T and CANPLACE(word, s, (dr, dc), G) then PLACE-
WORD(word, s, (dr, dc), G, Occupancy) PUSH(Stack, (word, s, (dr, dc))); T  $\leftarrow$ 
T  $\cup$  {key}
16:        return true
17:      end if
18:    end for
19:  end for
20:  ▷ Random placement with attempt cap (100 attempts)
21:  attempts  $\leftarrow$  0
22:  while attempts < 100 do
23:    (dr, dc)  $\leftarrow$  RANDOMCHOICE(D); s  $\leftarrow$  random start in G
24:    key  $\leftarrow$  (s, dr, dc); attempts  $\leftarrow$  attempts + 1
25:    if key  $\notin$  T and CANPLACE(word, s, (dr, dc), G) then PLACE-
WORD(word, s, (dr, dc), G, Occupancy) PUSH(Stack, (word, s, (dr, dc))); T  $\leftarrow$  T  $\cup$ 
{key}
26:    return true
27:  end if
28: end while
29: return false
30: end function

```

Algorithm 4 ASL-FINALIZE: Grid Completion

```

1: Input: G (partially filled grid)
2: Output: Completed grid G
3: function ASLFINALIZE(G)
4:   Replace remaining—in G by random letters
5:   return G
6: end function

```

Given an input list of words, the system first normalizes and deduplicates them, then assigns each word a “commonality” score based on how many letters it shares with other words. Words with more shared characters and greater length are placed earlier, since they are both more constrained and better candidates for intersections [27]. A random subset of words is then designated as diagonal, while the rest must be placed horizontally or vertically. This role assignment, combined with randomized direction choices, produces varied grids where a non-trivial fraction of words appear along diagonals, making the puzzle visually richer and slightly different from standard 4-direction word searches.

Backtracking is a standard tool in puzzle generation because naive greedy placement often leads to dead ends where remaining words cannot be inserted without overlaps or boundary violations. WoSeCon adopts a local and randomized variant of backtracking

rather than full recursive search. Words are processed sequentially, candidate placements are drawn from a global pool, and on failure the algorithm steps back to the previous word, restores its last placement to the pool, and retries untested locations in a randomized order. This yields a local, word-by-word backtracking process that explores many alternatives without explicitly enumerating the full combinatorial search tree [26].

In our implementation, we adopt the same idea but introduce an explicit cap on backtracking steps, maintain per-cell occupancy to safely undo only the last word, and add an intersection-first placement bias to produce fast, intersection-rich ASL word-search grids.

4. Evaluation and Results

We conducted comprehensive evaluations of both the isolated sign language recognition model and the puzzle generation algorithm to assess their technical performance, reliability, and suitability for educational applications. These technical evaluations employed quantitative metrics across multiple dimensions, providing insights into the system’s computational capabilities and limitations (see Section 4.1). Subsequently, we evaluated the system’s educational effectiveness through a human-subject study, measuring user experience, engagement, and learning outcomes (see Section 4.2). Together, these two complementary evaluations address both system-centric performance and human-centric pedagogical value.

4.1. Technical Evaluation

We first present the quantitative performance of the core technical components, beginning with the isolated sign language recognition model that enables real-time webcam-based word entry.

4.1.1. ISLR Model Performance

Our transformer-based model was trained for 500 epochs on the WLASL-100 subset, which comprises 100 common ASL signs selected for their pedagogical relevance and frequency of use. The training dynamics, visualized in Figure 10, reveal stable convergence patterns with a minor instability event occurring at epoch 326. This temporary divergence was effectively managed and corrected through the implementation of a cosine annealing schedule, which modulated the learning rate to facilitate recovery while preventing overshooting of optimal parameter values. The training trajectory demonstrates stable convergence. While the disparity between training (99.63%) and test (75.49%) Top-1 accuracy reflects a generalization gap typical of models trained on limited subsets like WLASL-100, the high Top-5 test accuracy (88.48%) indicates that the model successfully learns robust feature representations suitable for the game’s interactive context. Following the comprehensive evaluation standards established by recent pose-based transformer studies such as Sign2Pose [48], we report both Top-1 and Top-5 metrics to provide a complete picture of absolute predictive capability versus interactive robustness (see Table 3).

Table 3. ISLR Model Performance Metrics.

Metric	Value
Training Top-1 Accuracy	99.63%
Training Top-5 Accuracy	100.00%
Test Top-1 Accuracy	75.49%
Test Top-5 Accuracy	88.48%
Model Inference Latency (avg)	140.45 ms
End to end Latency (avg)	2.66 s
Model Size	11 MB

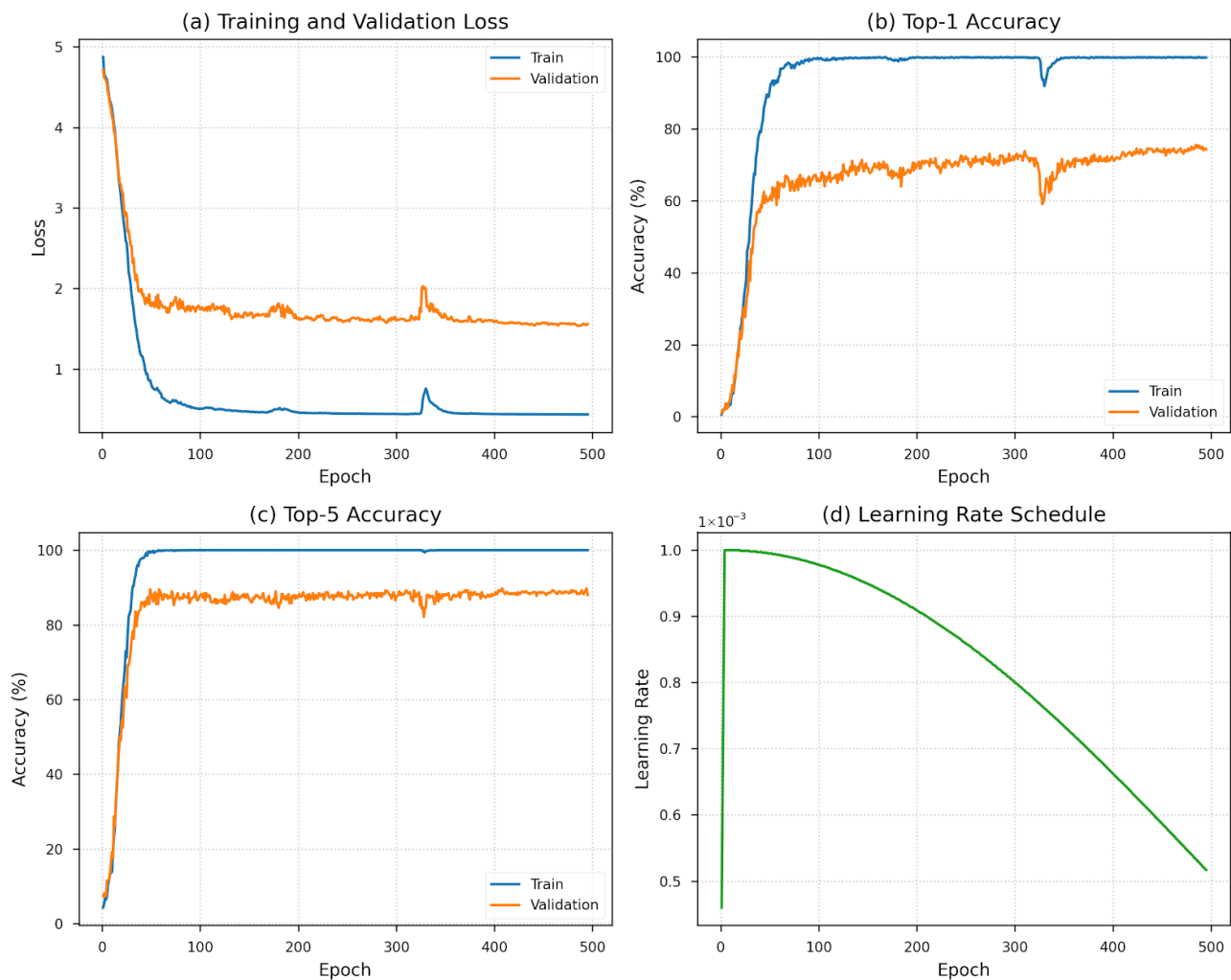


Figure 10. Training and validation metrics over 500 epochs.

Comparative analysis against established sign language recognition methods, detailed in Table 4, contextualizes our model's performance within the broader research landscape. Our approach achieves 75.49% Top-1 accuracy, positioning it between the 76.36% of SignBERT and the 82.95% of the state-of-the-art SSTAN method. This performance represents a substantial improvement over earlier graph-based approaches like Pose-GRU (46.51%) and Pose-TGCN (55.43%), demonstrating the effectiveness of transformer architectures for sign language recognition tasks. While our model does not achieve the highest absolute accuracy, it maintains significantly lower computational requirements than more complex alternatives, making it particularly suitable for deployment in resource-constrained educational environments where accessibility and responsiveness outweigh marginal accuracy gains.

Table 4. Comparison with State-of-the-Art ISLR Methods.

Method	Top-1 Acc	Top-5 Acc
Pose-GRU [18]	46.51%	76.74%
Pose-TGCN [18]	55.43%	78.68%
Sign2Pose [48]	80.9%	N/A
SignBERT [34]	76.36%	91.09%
SSTAN [35]	82.95%	95.35%
Our work	75.49%	88.48%

To better understand the model’s behavior across the 100-word vocabulary, we analyzed the accuracy distribution at the class level as shown in Table 5. The model demonstrates high consistency: 73% of all evaluated classes achieved a Top-1 accuracy of 75% or higher, with 34% of the vocabulary (including macro-movement signs like *basketball*, *table*, and *work*) achieving perfect 100% Top-1 accuracy. Only 7% of the classes exhibited a Top-1 accuracy below 50%.

Table 5. Accuracy Distribution Across Classes.

Top-1 Accuracy Range	Number of Classes	Percentage of Classes
100% (Perfect Prediction)	34	34%
75–99%	39	39%
50–74%	20	20%
<50%	7	7%

An analysis of the lowest-performing signs reveals that misclassifications primarily occur with signs featuring complex regional morphologies (e.g., *thanksgiving*) or highly localized positional subtleties (e.g., *man* and *woman*, which rely heavily on chin versus forehead placement). However, evaluating these “failed” predictions through the lens of Top-5 accuracy reveals significant model robustness. We saw that several signs that struggled in absolute Top-1 classification achieved near-perfect Top-5 accuracy.

To rigorously evaluate the system’s suitability for real-time interaction, we conducted latency benchmarking across the entire processing pipeline. All evaluations were performed on a machine equipped with an Apple M3 processor (8-core CPU, arm64 architecture) and 16.0 GB of total RAM, running macOS (version 26.2). This setup represents a standard, modern, CPU-only environment, aligning with our goal of deploying the application without requiring users to have proprietary hardware or dedicated GPUs.

The system was developed using Python v3.12 (Python Software Foundation, Wilmington, DE, USA), React v18.3.1 (Meta Platforms, Inc., Menlo Park, CA, USA), and PyTorch v2.8.0 (Meta AI, Menlo Park, CA, USA). The backend framework utilized Flask v3.0.0 with Flask-CORS v4.0.0 for cross-origin resource sharing, Python-Dotenv v1.0.0 for environment variable management, and Gunicorn v21.2.0 as the WSGI HTTP server. Additional libraries included NumPy v1.24.0 for numerical computations, OpenCV-Python-Headless v4.8.0 for image and video processing, MediaPipe v0.10.9 for real-time hand and pose tracking, FFmpeg-Python v0.2.0 for multimedia processing, and Pandas v2.0.0 for data manipulation and analysis.

Latency was measured over a validation set of 25 distinct sign language video sequences. To provide a transparent view of the system’s responsiveness, we isolated the computational cost of the core isolated sign language recognition (ISLR) model from the preliminary feature extraction steps. As detailed in Table 6, the average duration for the transformer model’s forward pass is 140.45 ms (± 9.83 ms), which validates our claim of low-latency inference suitable for real-time pedagogical feedback.

Table 6. Inference Latency Breakdown across Pipeline Stages (Measured over 25 samples).

Pipeline Step	Mean (ms)	Std. Dev (ms)	Min (ms)	Max (ms)
Video Read + MediaPipe Extraction	2508.15	349.14	1826.60	3214.05
Skeleton Preprocessing	15.04	1.03	12.57	18.16
Model Forward Pass	140.45	9.83	127.45	175.84
Total End-to-End Latency	2663.64	350.25	1978.90	3388.05

The initial extraction of skeletal landmarks via MediaPipe constitutes the bulk of the computational load (averaging 2508.15 ms), followed by a highly efficient skeleton preprocessing step (15.04 ms). By breaking down these metrics, we demonstrate that the recognition backbone itself is highly optimized, while the end-to-end latency remains within acceptable boundaries for asynchronous, game-based interaction. Also, once a user completes the puzzle, they get to see the time taken to complete and a notification of completion (see Figure 11).

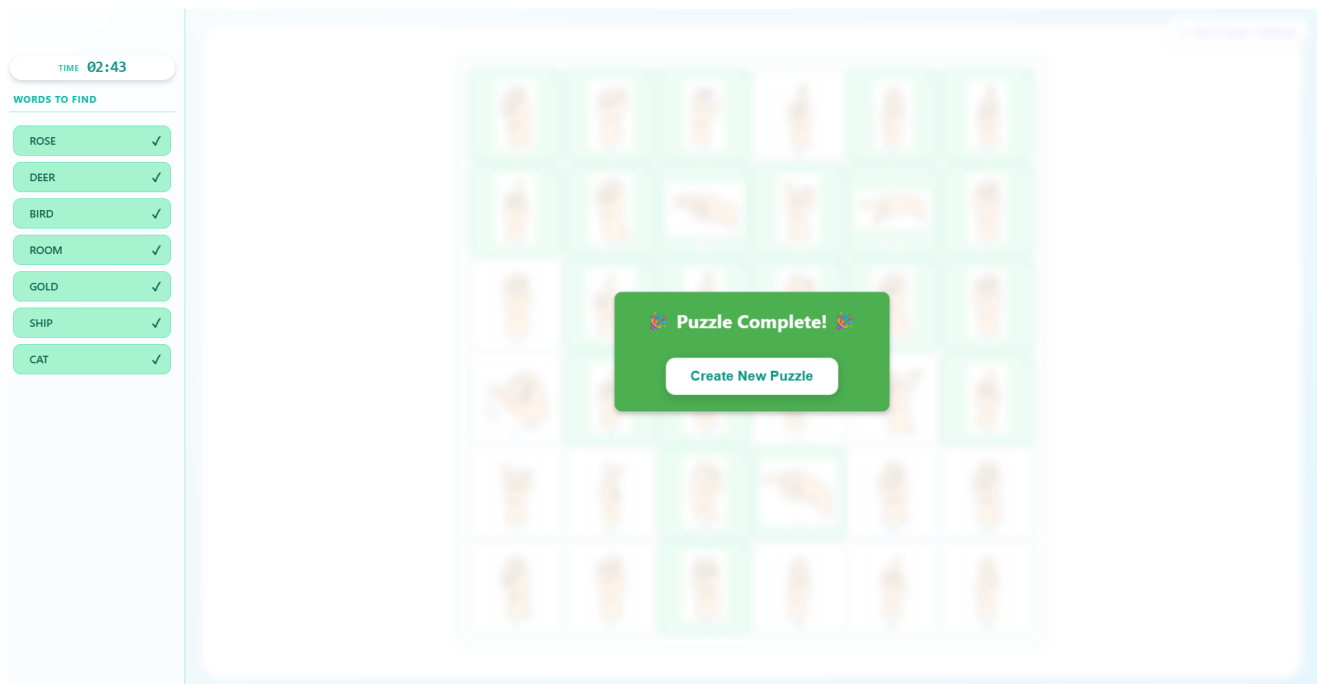


Figure 11. Puzzle completion screen. The interface displays the list of words the user has found and the total time taken to complete the game.

Integrating real-time machine learning into educational games necessitates a careful analysis of how system limitations—specifically misclassifications affect learner engagement. To mitigate the impact of recognition ambiguity, our system intentionally outputs Top-5 predictions rather than a single absolute classification. From a pedagogical perspective, when the system fails to rank the correct sign as the primary prediction, the user is presented with a multiple-choice disambiguation task. Rather than causing a hard failure that disrupts gameplay, this design translates AI uncertainty into a supplementary recognition exercise, requiring learners to visually identify their intended English word from a list of closely related alternatives.

4.1.2. Puzzle Generation Evaluation

The puzzle generation algorithm underwent systematic testing across 50 randomized trials with varying configurations to assess its reliability, efficiency, and output quality. Results presented in Table 7 demonstrate robust performance across all evaluated metrics.

The algorithm achieved a perfect 100% success rate in generating valid puzzles, confirming its reliability for educational applications where consistent performance is essential. Generation time scales appropriately with puzzle complexity, requiring only 0.085 s for 12×12 grids and 0.205 s for 20×20 grids, ensuring responsive user experience even for larger puzzles. The intersection utilization metric of 75.1% indicates efficient use of grid space while maintaining appropriate spacing between words, and the diagonal placement rate of 42.2% reflects the algorithm's flexibility in creating diverse and challenging puzzles.

zle configurations. The average backtracking count of 20.6 steps demonstrates efficient search through the solution space, avoiding exponential time complexity that would render real-time generation impractical.

Table 7. Puzzle Generation Performance Metrics.

Metric	Value
Success Rate	100%
Generation Time (12 × 12)	0.085 s
Generation Time (20 × 20)	0.205 s
Intersection Utilization	75.1%
Diagonal Placement Rate	42.2%
Backtracking Steps (avg)	20.6

The algorithm demonstrates particular strength in handling challenging edge cases, including word lists with zero character overlap where traditional placement approaches often fail. This capability ensures robust performance across diverse educational contexts where vocabulary lists may not share common letters or patterns. The consistent maintenance of real-time performance across all tested configurations confirms the algorithm’s suitability for interactive applications, where immediate feedback and rapid puzzle generation contribute significantly to user engagement and learning flow.

The puzzle generation algorithm operates with an expected time complexity of $O(V^2 \times L + B \times W \times H \times L)$, where V is vocabulary size, L is average word length, $W \times H$ represents grid dimensions, and B is the maximum backtracking budget. By strictly capping backtracking attempts, the algorithm avoids the exponential time complexity ($O(D^V)$) typical of naive recursive placement, maintaining a highly efficient space complexity of $O(W \times H + V)$. Empirically, the system scales exceptionally well under normal conditions: generating 25×25 or 40×40 grids with up to 100 words maintains a 100% success rate with near-instantaneous execution (averaging under 15 ms). When mathematically over-constrained—such as attempting to force 100 words into a 15×15 grid—the algorithm safely utilizes its backtracking limits. In these edge cases, generation time temporarily increases to approximately 900 ms and success rates naturally drop, but the strict boundary prevents the system from crashing or hanging.

Compared to existing generation techniques like standard unconstrained crossword strategies, our approach is uniquely optimized for interactive, real-time web environments. It introduces an intersection-first heuristic that explicitly supports diagonal placements, a feature frequently omitted in standard generators. Most importantly, while traditional exhaustive-search algorithms can stall indefinitely when a valid grid cannot be resolved, our bounded backtracking guarantees swift termination. This design choice prioritizes the user experience, ensuring that learners receive immediate feedback and visually rich, interconnected puzzles without suffering from backend computational bottlenecks.

4.1.3. Summary of Technical Findings

Here are the highlights of the technical findings:

- ISLR Top-5 accuracy: 88.48%
- Model inference latency: 140.45 ms
- Puzzle generation success rate: 100%
- Generation time (20 × 20): 0.205 s

The technical results above confirm system viability. We now evaluate whether these capabilities translate into effective learning outcomes.

4.2. User Study Evaluation

Having established the technical viability of the ISLR model and puzzle generation algorithm, we next evaluate the system's educational effectiveness and user experience through a human-subject study.

4.2.1. User Study Results

The evaluation of the “ASL Word Puzzle” application involved 37 participants (21 female, 16 male), all between the ages of 19 and 25. All participants were enrolled in “ASL 101: Introduction to ASL” and had beginner-level familiarity with American Sign Language. The study was conducted in a university classroom setting. Two administrators helped facilitate data collection, briefing, question answering, and demonstration of the tool. Participants sat in a classroom, each with their own computer. They were first briefed on the game and shown how to use its basic functionalities. However, most were able to explore and navigate the application independently, likely due to their familiarity with technology. Overall, participants appeared comfortable using the game.

The evaluation was conducted in two rounds: 25 participants took part in the first round and 12 in the second round, with no overlap between participants in the two sessions (each session was about 45 min). The analysis focused on three main dimensions: User Experience (UX), Learning Outcomes, and a comparative analysis of self-reported skills before and after the activity. Each 45-min session begins with a 5-min briefing and pre-activity self-assessment, followed by 25 min of gameplay split into two rounds (free exploration then fixed puzzles). A 5-min check-in halfway through gathers quick UX feedback, and the session ends with a 5-min post-activity self-assessment and wrap-up. This structure supports analysis of user experience, learning outcomes, and skill improvement before vs. after the activity.

4.2.2. Quantitative Analysis

Table 8 summarizes the mean scores and standard deviations for the Likert-scale questions (1 = Strongly Disagree, 5 = Strongly Agree) collected from 37 participants. Participants reported high levels of satisfaction across all usability metrics, with the highest engagement score attributed to the puzzle-based format ($M = 4.49$, $SD = 0.51$). Notably, users strongly agreed that the puzzle format aided memory retention better than traditional video resources ($M = 4.54$, $SD = 0.61$).

Table 8. User Experience and Learning Evaluation Scores ($N = 37$).

Category	Question Item	Mean	SD
Usability	Instructions for each puzzle were clear	4.30	0.62
	Overall navigation was intuitive	4.24	0.76
	Easy to understand how to start the game	4.14	0.79
	Clarity of the next steps in the puzzle	4.35	0.54
Design	ASL signs were clear and easy to see	4.32	0.58
	Visual design helped focus on learning	4.16	0.65
	Colors, icons, and cues were easy to understand	4.27	0.56
	The puzzle-based format was engaging	4.49	0.51
Learning	This activity improved my ASL vocabulary	4.27	0.69
	Better than videos alone for remembering signs	4.54	0.61
	This game would be effective for ASL beginners	4.08	0.64

The pre- and post-activity comparison (Table 9) revealed a significant increase in participant confidence. A paired *t*-test showed that confidence in learning ASL independently

improved significantly from $M = 3.51$ to $M = 4.14$ ($p < 0.001$). While the self-reported ability to recognize ASL words also increased from $M = 3.86$ to $M = 4.00$, this change was not statistically significant ($p = 0.164$), likely due to the high baseline familiarity reported by participants ($M = 4.00$).

Table 9. Pre- and Post-Activity Self-Evaluation ($N = 37$).

Metric	Pre-Activity Mean	Post-Activity Mean	<i>p</i> -Value
Ability to recognize ASL words	3.86	4.00	0.164
Confidence in independent learning	3.51	4.14	<0.001 ***

*** Significant at $p < 0.001$.

The results demonstrate that integrating real-time Isolated Sign Language Recognition (ISLR) into a gamified word-search framework is both technically viable and pedagogically effective.

4.2.3. Usability and Engagement

The high scores in usability ($M \geq 4.14$) suggest that the browser-based interface successfully minimized the technical barriers often associated with webcam-based recognition. The engagement score for the puzzle format ($M = 4.49$) highlights the value of gamification in maintaining learner motivation. This supports the hypothesis that interactive mechanics can transform the often-passive experience of watching sign language videos into an active linguistic practice.

4.2.4. Pedagogical Impact

A key finding is the perceived superiority of the puzzle format over static video resources for memory retention ($M = 4.54$). By requiring users to identify signs within a competitive or exploratory context (the word search), the system encourages deeper cognitive processing. The significant boost in confidence for independent learning ($p < 0.001$) suggests that the immediate feedback provided by the recognition model empowers users, reducing the anxiety often associated with learning a spatial language without a live instructor.

4.2.5. Statistical Power and Error Analysis

The statistical validity of our findings is further reinforced by an analysis of the Type II error probability (β). A Type II error occurs when the null hypothesis is incorrectly retained despite the presence of a true experimental effect. The relationship between the probability of committing a Type II error and the statistical power ($1 - \beta$) is defined by the following equation:

$$\beta = 1 - \text{Power} = P(\text{Type II Error}) \quad (1)$$

For our primary metric, participant confidence, the post-hoc power analysis yielded a value of 0.999. Applying the equation above, the probability of a Type II error for this result is $\beta < 0.001$, which is substantially lower than the maximum threshold of $\beta = 0.20$ typically recommended for behavioral research [49]. This indicates that our sample size of $N = 37$ was more than sufficient to reliably detect the significant improvements in learner confidence. Conversely, for the word recognition ability metric, the higher observed β of 0.718 suggests that while the current study provides preliminary evidence, it may be underpowered for detecting smaller effect sizes ($d = 0.23$), a common characteristic of pilot interventions [50].

4.2.6. Summary of User Study Findings

- Significant increase in learner confidence ($p < 0.001$)
- High usability scores ($M \geq 4.14$ out of 5)
- Strong perceived learning benefit ($M = 4.54$ for retention vs. videos)

Together with the technical results above, these findings demonstrate that the system is both computationally viable and educationally effective.

4.2.7. Future Improvements

Qualitative feedback, though limited, indicated a desire for more diverse vocabulary and multiple-choice modalities to complement the recognition tasks. Future iterations should focus on expanding the word list and refining the recognition accuracy for more complex signs, as suggested by users who noted the importance of video clarity and accuracy in professional educational contexts.

5. Discussion and Conclusions

Our research presents a novel approach to American Sign Language education through the development of an interactive ASL Word Search Game that combines real-time sign language recognition with engaging educational game mechanics. The system successfully demonstrates that computer vision technologies, particularly those inspired by recent advances in bidirectional ASL-English translation systems [21], can be effectively integrated into accessible learning platforms. The resulting application provides an intuitive and visually-oriented environment for vocabulary practice that aligns with the natural learning preferences of Deaf and Hard of Hearing users, while also serving as an effective tool for hearing individuals learning ASL.

The experimental evaluation reveals several important findings regarding the system's performance and educational value. First, we demonstrate that lightweight isolated sign language recognition models can achieve practical accuracy levels suitable for educational applications, with our implementation reaching 88.48% Top-5 accuracy while maintaining real-time performance in web browser environments. This performance level, though not perfect, proves sufficient for vocabulary reinforcement activities where the educational context provides additional cues for sign recognition. Second, our constraint-based puzzle generation algorithm reliably produces challenging yet solvable grids that reinforce spatial relationships between signs, effectively supporting vocabulary acquisition through pattern recognition and visual search processes. Third, the implementation of multiple input modalities, including webcam-based sign recognition, manual word entry, file upload, and an interactive sign builder, significantly enhances accessibility by accommodating diverse user preferences, technological constraints, and learning styles.

The educational implications of this work extend beyond the specific game mechanics. By integrating sign recognition capabilities directly into the puzzle interface, we create a closed-loop learning system where users can both input signs through webcam recognition and then search for those same signs within the puzzle context. This bidirectional interaction mirrors the natural language learning process where production and recognition skills develop interdependently. The system's dual learning modes, Beginner with English letter hints and Advanced with only ASL signs, provide appropriate scaffolding for learners at different proficiency levels, supporting progressive skill development.

While the current system evaluates the WLASL-100 dataset, our underlying skeletal extraction and Transformer architecture is vocabulary-agnostic and scales natively to the full WLASL-2000 dataset without structural changes. However, scaling to 2000 classes involves practical trade-offs. Training on this exponentially larger dataset requires substantial computational resources, necessitating a multi-GPU configuration to complete training in

a reasonable timeframe. Additionally, we anticipate a moderate degradation in baseline Top-1 accuracy due to increased inter-class visual similarity.

Several limitations of the current implementation warrant acknowledgment and suggest directions for future research. The vocabulary size constraint of 100 words, while sufficient for proof-of-concept demonstration, limits the system's utility for advanced learners. Additionally, the webcam recognition component remains somewhat dependent on consistent lighting conditions and camera quality, which may affect accessibility for users with limited technological resources. The current puzzle generation algorithm, while effective for small to medium grid sizes, may require optimization for larger, more complex puzzles with extensive vocabulary sets.

Future work will focus on addressing these limitations while expanding the system's capabilities. We plan to explore transfer learning techniques to significantly expand the recognizer's vocabulary while maintaining real-time performance. This expansion will enable the system to support more comprehensive ASL curricula and accommodate learners at advanced proficiency levels. We also intend to investigate multi-user collaborative gameplay modes, recognizing that language learning often occurs most effectively in social contexts. The development of adaptive difficulty algorithms based on real-time assessment of user performance represents another promising direction, potentially personalizing the learning experience to individual skill levels and learning trajectories. Finally, formal integration with established ASL curricula and educational standards will be essential for the system's adoption in formal educational settings, requiring collaboration with ASL educators and curriculum developers.

Beyond these technical improvements, the system represents a significant conceptual contribution to the field of accessible educational technology. By successfully bridging computer vision, educational game design, and accessibility principles, we provide a model for future developments in inclusive learning tools. The demonstration that real-time sign recognition can be effectively deployed in browser-based environments without specialized hardware lowers barriers to access and democratizes ASL education technology. This accessibility aligns with the broader goal of making language learning resources available to diverse populations regardless of socioeconomic status or geographical location.

In conclusion, the ASL Word Search Game represents a meaningful step forward in interactive ASL education technology. By combining validated educational principles with state-of-the-art computer vision techniques [21], we have created a system that not only teaches ASL vocabulary but does so through an engaging, visually-oriented interface that respects the visual nature of sign language. The positive user feedback and promising experimental results suggest that this approach has significant potential for enhancing ASL learning outcomes. As sign language recognition technology continues to advance, particularly through innovations in bidirectional translation systems, we anticipate increasingly sophisticated and effective educational applications that will further support ASL acquisition and promote greater understanding between Deaf and hearing communities.

Author Contributions: Conceptualization: S.A. and M.A.; methodology: S.A. and M.A.; software: R.M.B.; validation: S.A., M.A., R.M.B., L.N. and M.A.A.D.; formal analysis: S.A., M.A. and M.A.A.D.; investigation: S.A. and M.A.; resources: S.A., M.A. and L.N.; data curation: S.A. and R.M.B.; writing—original draft preparation: S.A., M.A. and R.M.B.; writing—review and editing: S.A., M.A., R.M.B. and M.A.A.D.; visualization: S.A., M.A. and R.M.B.; supervision: M.A. and S.A.; project administration: M.A. and S.A.; funding acquisition: M.A., S.A. and M.A.A.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partly funded by the Pennsylvania State System of Higher Education (PASSHE) Faculty Professional Development Council (FPDC) grant and Provost Research Grant (PRG) of West Chester University.

Data Availability Statement: The dataset used in this study is publicly available at: <https://dxli94.github.io/WLASL/> (accessed on 15 January 2026).

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

References

1. Aljedaani, W.; Krasniqi, R.; Aljedaani, S.; Mkaouer, M.W.; Ludi, S.; Al-Raddah, K. If online learning works for you, what about deaf students? Emerging challenges of online learning for deaf and hearing-impaired students during COVID-19: A literature review. *Univers. Access Inf. Soc.* **2023**, *22*, 1027–1046. [CrossRef]
2. Amiruzzaman, S.; Amiruzzaman, M.; Begum, H.; Bhati, D.; Wu, T.H. Deepening Mathematical Understanding Using Visualization and Interactive Learning for Deaf Students. *Eng. Proc.* **2025**, *89*, 4.
3. Korte, J. Critical analysis of the design of online learning environments for deaf and hard of hearing students. *J. Spec. Educ. Apprenticesh.* **2020**, *9*, n2.
4. Mitchell, R.E. Quantifying the benefit of early intervention for deaf and hard-of-hearing children: Findings from the Gallaudet Research Institute. *Infants Young Child.* **2006**, *19*, 200–211.
5. Bragg, D.; Koller, O.; Bellard, M.; Berke, L.; Boudreault, P.; Braffort, A.; Caselli, N.; Huenerfauth, M.; Kacori, H.; Verhoef, T.; et al. Sign language recognition, generation, and translation: An interdisciplinary perspective. In Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility, Pittsburgh, PA, USA, 28–30 October 2019; pp. 16–31.
6. May, J.; Brennan, K.; Amiruzzaman, S.; Amiruzzaman, M. English to American Sign Language: An AI-Based Approach. *J. Comput. Sci. Coll.* **2024**, *40*, 164–175.
7. Chen, X.; Zhang, Y.; Du, S. A systematic review of online learning for deaf and hard of hearing students. *Educ. Inf. Technol.* **2022**, *27*, 4891–4915.
8. Hoffmeister, R.J.; Caldwell-Harris, C.L. Acquiring English as a second language via print: The task for deaf children. *Cognition* **2013**, *132*, 229–242. [CrossRef]
9. Vygotsky, L.S. *Mind in Society: The Development of Higher Psychological Processes*; Harvard University Press: Cambridge, MA, USA, 1978.
10. Engelkamp, J. *Memory for Actions*; Psychology Press: East Sussex, UK, 1998.
11. Brashear, H.; Henderson, V.; Park, K.H.; Hamilton, H.; Lee, S.; Starner, T. American sign language recognition in game development for deaf children. In Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility, Portland, OR, USA, 23–25 October 2006; pp. 79–86.
12. Hung, C.M.; Huang, I.; Hwang, G.J. Effects of digital game-based learning on students' self-efficacy, motivation, anxiety, and achievements in learning mathematics. *J. Comput. Educ.* **2014**, *1*, 151–166. [CrossRef]
13. Lubis, L.S. The Effect of Word Search Puzzle Game to Vocabulary Mastery at the VIII Grade Students of SMP N 5 Padangsidempuan. Ph.D. Thesis, IAIN Padangsidempuan, Padang Sidempuan, Indonesia, 2021.
14. Wahyudi, M.A. The Effectiveness of Word Search Puzzle Game as Media for Teaching Vocabulary. *ELP (J. Engl. Lang. Pedagog.)* **2024**, *9*, 43–52. [CrossRef]
15. Plass, J.L.; Homer, B.D.; Kinzer, C.K. Foundations of game-based learning. In *Handbook of Game-Based Learning*; MIT Press: Cambridge, MA, USA, 2015; pp. 3–24.
16. Gafoor, K.A.; Umer, F. Effectiveness of word search puzzles in enhancing vocabulary among secondary school students. *J. Educ. Res.* **2012**, *15*, 67–74.
17. Medronha, A.; Lima, L.; Claudio, J.; Kupssinskü, L.; Barros, R.C. LERMO: A novel web game for AI-enhanced sign language recognition. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 23352–23359. [CrossRef]
18. Li, D.; Rodriguez, C.; Yu, X.; Li, H. Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 1459–1469.
19. Yan, S.; Xiong, Y.; Lin, D. Spatial temporal graph convolutional networks for skeleton-based action recognition. *Proc. AAAI Conf. Artif. Intell.* **2018**, *32*, 7444–7452. [CrossRef]
20. Boháček, M.; Hruží, M. Sign pose-based transformer for word-level sign language recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 4–8 January 2022; pp. 182–191.
21. Amiruzzaman, S.; Amiruzzaman, M.; Batchu, R.M.; Dracup, J.; Pham, A.; Crocker, B.; Ngo, L.; Dewan, M.A.A. Bidirectional Translation of ASL and English Using Machine Vision and CNN and Transformer Networks. *Computers* **2026**, *15*, 20. [CrossRef]
22. Vijitkunsawat, W.; Racharak, T. GSR-Fusion: A Deep Multimodal Fusion Architecture for Robust Sign Language Recognition Using RGB, Skeleton, and Graph-Based Modalities. *IEEE Access* **2025**, *13*, 108235–108254. [CrossRef]
23. Nielsen, J. *Usability Engineering*; Morgan Kaufmann: San Francisco, CA, USA, 1993.

24. Agarwal, C.; Joshi, R.K. Automation strategies for unconstrained crossword puzzle generation. *arXiv* **2020**, arXiv:2007.04663. [CrossRef]
25. Moussiades, L. A Word Search Puzzle Construction Algorithm. *Int. J. Sci. Eng. Sci.* **2022**, *6*, 121–126.
26. Balogun, G.B.; Ibisagba, D.; Bajeh, A.; Debo, T.O.; Muyeideen, A.; Peter, O.J. Comparative analysis of AI-based search algorithms in solving 8 puzzle problems. *Bull. Natl. Res. Cent.* **2024**, *48*, 119. [CrossRef]
27. Arsov, D.; Kitanovski, T.; Jovanov, M. Crossword Generation as a Constraint Satisfaction Problem Using Parallel Processing and Lemmatization. In *ICT Innovations 2024. TechConvergence: AI, Business, and Startup Synergy*; Risteska Stojkoska, B., Janeska Sarkanjac, S., Eds.; Springer Nature: Cham, Switzerland, 2025; pp. 34–48. [CrossRef]
28. Chebka, R.; Essalmi, F. A crosswords game for deaf. In *Proceedings of the 2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA)*; IEEE: Piscataway, NJ, USA, 2015; pp. 1–6.
29. Avina, V.D.; Amiruzzaman, M.; Amiruzzaman, S.; Ngo, L.B.; Dewan, M.A.A. An AI-Based Framework for Translating American Sign Language to English and Vice Versa. *Information* **2023**, *14*, 569. [CrossRef]
30. Locke, J.L. A theory of neurolinguistic development. *Brain Lang.* **1997**, *58*, 265–326. [CrossRef]
31. Meier, R.P.; Newport, E.L. Out of the hands of babes: On a possible sign advantage in language acquisition. *Language* **1990**, *66*, 1–23. [CrossRef]
32. Padden, C.; Ramsey, C. American Sign Language and reading ability in deaf children. In *Language Acquisition by Eye*; Psychology Press: Hove, UK, 1999; pp. 165–189.
33. Bragg, D.; Caselli, N.; Gallagher, J.W.; Goldberg, M.; Oka, C.J.; Thies, W. ASL sea battle: Gamifying sign language data collection. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, Virtual, 8–13 May 2021; pp. 1–13.
34. Hu, H.; Zhao, W.; Zhou, W.; Li, H. SignBERT+: Hand-Model-Aware Self-Supervised Pre-Training for Sign Language Understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 11221–11239. [CrossRef] [PubMed]
35. Hirooka, K.; Miah, A.S.M.; Murakami, T.; Hasan, M.A.M.; Hwang, Y.S.; Shin, J. Stack Transformer Based Spatial-Temporal Attention Model for Dynamic Sign Language and Fingerspelling Recognition. *arXiv* **2025**, arXiv:2503.16855. [CrossRef]
36. Hu, H.; Zhou, W.; Pu, J.; Li, H. Global-local Enhancement Network for NMFs-aware Sign Language Recognition. *arXiv* **2020**, arXiv:2008.10428. [CrossRef]
37. Saunders, B.; Camgöz, N.C.; Bowden, R. Progressive Transformers for End-to-End Sign Language Production. *arXiv* **2020**, arXiv:2004.14874.
38. Joze, H.R.V.; Koller, O. MS-ASL: A Large-Scale Data Set and Benchmark for Understanding American Sign Language. *arXiv* **2018**, arXiv:1812.01053.
39. Desai, A.; Berger, L.; Minakov, F.O.; Milan, V.; Singh, C.; Pumphrey, K.; Ladner, R.E.; Daumé, H., III; Lu, A.X.; Caselli, N.; et al. ASL Citizen: A Community-Sourced Dataset for Advancing Isolated Sign Language Recognition. *arXiv* **2023**, arXiv:2304.05934. [CrossRef]
40. Forster, J.; Schmidt, C.; Koller, O.; Bellgardt, M.; Ney, H. Extensions of the Sign Language Recognition and Translation Corpus RWTH-PHOENIX-Weather. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*; Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S., Eds.; European Language Resources Association (ELRA): Reykjavik, Iceland, 2014; pp. 1911–1916.
41. Zhou, H.; Zhou, W.; Qi, W.; Pu, J.; Li, H. Improving Sign Language Translation with Monolingual Data by Sign Back-Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual, 19–25 June 2021; pp. 131–140.
42. Liu, Z.; Pang, L.; Qi, X. MEN: Mutual Enhancement Networks for Sign Language Recognition and Education. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 311–325. [CrossRef]
43. Duarte, A.C.; Palaskar, S.; Ghadiyaram, D.; DeHaan, K.; Metze, F.; Torres, J.; Giró-i-Nieto, X. How2Sign: A Large-scale Multimodal Dataset for Continuous American Sign Language. *arXiv* **2020**, arXiv:2008.08143.
44. Cao, Z.; Simon, T.; Wei, S.; Sheikh, Y. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *arXiv* **2016**, arXiv:1611.08050.
45. Grishchenko, I.; Bazarevsky, V. MediaPipe Holistic: Simultaneous Face, Hand and Pose Prediction on Device. 2020. Available online: <https://research.google/blog/mediapipe-holistic-simultaneous-face-hand-and-pose-prediction-on-device/> (accessed on 15 January 2026).
46. Jiang, M.; Dong, J.; Ma, D.; Sun, J.; He, J.; Lang, L. Inception Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In *2022 International Symposium on Control Engineering and Robotics (ISCEER)*; IEEE: Piscataway, NJ, USA, 2022; pp. 208–213. [CrossRef]
47. Wei, F.; Chen, Y. Improving Continuous Sign Language Recognition with Cross-Lingual Signs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2023; pp. 23612–23621.
48. Eunice, J.; J, A.; Sei, Y.; Hemanth, D.J. Sign2Pose: A Pose-Based Approach for Gloss Prediction Using a Transformer Model. *Sensors* **2023**, *23*, 2853. [CrossRef]

49. Cohen, J. A power primer. *Psychol. Bull.* **1992**, *112*, 155–159. [[CrossRef](#)] [[PubMed](#)]
50. Faul, F.; Erdfelder, E.; Lang, A.G.; Buchner, A. G*Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behav. Res. Methods* **2007**, *39*, 175–191. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.