Whisper, Translate, Speak, Sync: Video Translation for Multilingual Video Conferencing Using Generative AI

Amirkia Rafiei Oskooei[1(✉)], Eren Caglar[2], Ibrahim Şahin[1], Ayse Kayabay[1], and Mehmet S. Aktas[1]

[1] Department of Computer Engineering, Yildiz Technical University, Istanbul, Turkey
{amirkia.oskooei,ibrahim.sahin1,ayse.kayabay}@std.yildiz.edu.tr, aktas@yildiz.edu.tr

[2] Aktif Bank, R&D Center, Istanbul, Turkey
eren.caglar@aktifbank.com.tr
https://www.aktifbank.com.tr/

Abstract. This paper addresses the growing need for seamless communication in multilingual video conferencing by presenting a novel, computationally efficient methodology for real-time video translation. While advancements in neural networks have enabled accurate speech translation and voice cloning, integrating these with lip synchronization for realistic talking head generation remains a challenge, particularly for real-time applications. This paper introduces a comprehensive video translation pipeline leveraging open-source deep learning models. We further propose a scalable system architecture incorporating a "Token Ring" mechanism to manage speaker turns and minimize computational load, addressing key challenges related to latency, scalability, and personalization in multilingual settings. A segmented batched processing protocol with inverse throughput thresholding and overlapping buffering is implemented to achieve near real-time performance. A simplified, universal prototype is developed to demonstrate the feasibility and efficacy of our approach, providing a foundation for building next-generation multilingual video conferencing systems. This work offers a practical framework for developers and businesses aiming to create inclusive and effective communication platforms.

Keywords: Video Translation · Computer Vision · Deep Learning · Generative AI · Human-AI Interaction · Video Conferencing

1 Introduction

In today's digital world, communication is increasingly taking place in online environments. With global collaborations on the rise, the need for technology that can overcome language barriers has become a pressing challenge. Video

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2025 Workshops, LNCS 15890, pp. 217–234, 2026.
https://doi.org/10.1007/978-3-031-97606-3_15

conferencing has emerged as a cornerstone of modern communication, widely used in education, business, and work settings. The recent COVID-19 pandemic further accelerated the adoption of these platforms, enabling people to work and study remotely from home. When people with different native languages interact, challenges arise. Expressing oneself in a foreign language can be difficult, and understanding someone speaking in a non-native language can slow down and complicate the conversation. In an interconnected world, seamless communication across linguistic and cultural boundaries is essential, which drives the demand for real-time, multilingual video conferencing systems.

Recent advances in neural networks and deep learning have enabled rapid and accurate translation of both written and spoken language in virtual environments. Modern neural models are now capable of translating text and speech between multiple languages while preserving the tone and style of the speaker through voice cloning techniques. However, while these models can translate speech and adjust voice characteristics, the video component still presents a challenge. In video conferences, even if the translated speech sounds natural, the speaker's lip movements and facial expressions remain in the source language. Recent progress in lip synchronization, or audio-driven talking head generation, now allows for the generation of video where the speaker's lip movements match the translated speech. In simple terms, this means that a speaker's lips can be effectively "translated" along with their words. The complete process of translating all aspects of communication—including language, voice tone, facial expressions, and lip movements—is often called "Video Translation" or "Face-to-Face Translation" or simply "Face Translation".

Video Translation concept is gaining significant momentum as a potentially revolutionary communication paradigm. Its broad applicability spans diverse sectors including entertainment through video dubbing, film translation, immersive Augmented and Virtual Reality (AR/VR) experiences, telemedicine [6], and, crucially, multilingual video conferencing. Despite the acknowledged potential and numerous applications highlighted in current research, concrete methodologies for effectively deploying video-translation pipelines within real-world video conferencing systems remain underdeveloped. Furthermore, the practical challenges inherent in real-world implementation, and strategies to effectively address these challenges, require in-depth exploration. A primary obstacle lies in the computational demands of video-translation pipelines. The integration of multiple deep learning models, particularly computationally intensive generative models for computer vision tasks like lip synchronization, necessitates substantial processing resources, often requiring high-performance GPUs. Moreover, the imperative for real-time and uninterrupted communication in video conferencing environments exacerbates these computational pressures. Simply scaling computational resources by adding more or more powerful GPUs presents scalability concerns, especially in large multilingual meetings where numerous participants require real-time translation into their respective native languages, potentially demanding parallel execution of the entire pipeline multiple times.

Currently, a detailed exploration of these practical challenges is lacking in the existing literature, leaving a significant gap in actionable guidelines for devel-

opers and businesses. In response to these challenges, this paper presents a novel methodology that is computationally cost-effective, universal, and easy to implement. Our approach is designed to integrate with any video conferencing client while delivering near real-time performance. We propose a framework that includes a comprehensive video translation pipeline, a new system architecture, and strategies to address the computational and scalability challenges that have so far limited real-world applications. This work contributes to the literature by guiding developers and businesses toward building next-generation communication systems that enable seamless, multilingual interaction.

The subsequent sections of this paper are organized as follows: Sect. 2 provides a review of the current, albeit limited, literature on neural models and lip synchronization techniques applied to Video-translation. Section 3 details our proposed methodology, delineating the identified challenges and presenting our pipeline and system architecture as solutions. Section 4 describes the development of a universal and simplified prototype designed to validate the feasibility and efficacy of our approach. Finally, Sect. 5 presents a comprehensive evaluation of our system, encompassing both objective performance metrics and subjective user assessments, followed by an in-depth analysis of the results obtained.

2 Related Works

Speech-to-Speech Translation (S2S) has emerged as a pivotal technology for breaking down communication barriers in a globalized world. At its core, S2S systems typically involve a sequential pipeline comprising Automatic Speech Recognition (ASR), Machine Translation (MT), and Speech Synthesis (SS) modules. ASR transcribes spoken language into text, which is then translated into a target language by the MT module, and finally, the translated text is converted back to speech using SS. The advent of neural networks has revolutionized each of these components, leading to significant improvements in accuracy and naturalness compared to traditional statistical methods. Neural ASR models, often employing architectures like recurrent neural networks and transformers, have achieved remarkable performance in transcribing diverse accents and noisy environments [9,15,20]. Similarly, Neural Machine Translation (NMT) has surpassed phrase-based statistical MT in fluency and translation quality, leveraging encoder-decoder architectures and attention mechanisms [31,34]. More recently, the rise of large language models (LLMs) and their cpabilities in various tasks such as language and code understanding, has further pushed the boundaries of MT and even end-to-end S2S, demonstrating the potential for more contextually aware and human-like translations [19,40]. Neural Text-to-Speech (TTS) systems have also significantly improved the naturalness and expressiveness of synthesized speech, often incorporating voice cloning or voice conversion techniques to maintain speaker identity or style [4,13]. These advancements in neural models and LLMs have collectively propelled S2S towards more seamless and practical applications. The concept of Video Translation, sometimes referred to as Face Translation or Face-to-Face Translation, extends the capabilities of

S2S by addressing the visual aspects of communication, particularly in video settings. This area, while nascent compared to traditional S2S, is gaining increasing attention due to its potential to enhance video conferencing, film dubbing, and immersive experiences.

Extending S2S to the visual domain requires language-independent Lip Sync or Talking Head Generation models. This language independence is essential for integrating these models into video translation pipelines. Recent advances in neural language-independent Lip Sync models have been key to progress in face-to-face translation, shifting focus primarily to the lip sync module itself. A foundational work in language-independent lip sync using neural models was presented in [11]. This work spurred research into GAN-based lip sync models [3,7,8,10,21,33,39]. Subsequent research has explored Diffusion-based [12,14, 26,29,35] and NeRF-based [27,30,36–38] approaches. Some studies have also examined the generalizability and language independence of these models [23], a crucial requirement for video translation pipelines.

Having covered the necessary modules and models for Video Translation, we now focus on works specifically utilizing lip sync and S2S pipelines for this task. This area remains underexplored, but we highlight some prominent examples. One of the earliest attempts at Video Translation was [24], which developed a full face-translation pipeline before the advent of neural models. The [11] discussed the concept of automatic face-to-face translation and its applications, but did not provide specific guidance for real-world scenarios. The [28] integrates Multilingual TTS with Lip Sync, but lacks focus on real-world application and comprehensive experimental evaluation. [5] presents a notable effort in end-to-end face translation, developing a model that avoids cascading modules. However, this end-to-end approach, while advantageous in some ways, lacks modularity, which can limit control over individual steps and raise scalability concerns. The [22] implements a blazing-fast real-time pipeline and addresses concerns regarding latency, but does not address the system-level challenges.

While speech and video translation components have advanced significantly, deploying and evaluating complete video translation pipelines for real-world video conferencing remains a challenge. Existing work often focuses on individual modules like lip synchronization, neglecting system-level integration issues for real-time, scalable, and user-friendly applications. Computational demands and architectural considerations, especially for multilingual settings, are also under-explored. This paper addresses these gaps by presenting a novel, cost-effective, and universally applicable methodology for video translation tailored for multilingual video conferencing. We propose a complete pipeline, a scalable architecture, and practical strategies to overcome computational and scalability bottlenecks that have limited the adoption of video translation in real-world communication.

The authors of [1] proposed a fault-tolerant information service architecture for managing dynamic Web and Grid resources. The authors of [25] conducted a systematic mapping study demonstrating structured methodologies for evaluating application reliability. Research efforts in [17,18] modeled user behavior

through embedding-based representations of clickstream sequences, while the authors of [32] introduced a provenance-enabled data infrastructure to improve transparency in scientific workflows. In parallel, the authors of [2,16] addressed collaborative grid portals for high-performance computing applications. While these studies emphasize distributed system architectures, user modeling, and provenance integration, none of them address the real-time orchestration of generative AI models for speech, translation, and video synchronization in multilingual communication.

3 Methodology

In this section, we detail our methodology for designing a video translation pipeline utilizing deep learning models specifically for video conferencing systems. Our methodology prioritizes minimal computational resource and budgetary demands, aiming to facilitate the development of multilingual video conferencing systems that are cost-effective, universally accessible, and scalable. In the Subsect. 3.1, we describe the proposed pipeline workflow, which integrates readily available open-source deep learning models. Considering the application of this pipeline within video conferencing environments, the Subsect. 3.2 presents a scalable system architecture that leverages the proposed pipeline.

3.1 Pipeline

Our video translation pipeline leverages state-of-the-art open-source deep learning models, selected for their efficacy and broad accessibility. From a high-level perspective, the objective of this modular pipeline is to process an input video and generate a talking head of a specified identity—either the original individual or a novel identity—where the original speech is translated into a target language while preserving the original voice tone, and the lip movements are synchronized with the translated speech. This pipeline constitutes a deep learning-based video translation system, also referred to in the literature as Face Translation or Face-to-Face translation, designed for portable and versatile deployment.

We now detail the components and modules of our proposed pipeline, as depicted in Fig. 1. The input is a video featuring a talking head, or more generally, any speaking individual. The initial module in the pipeline is an **Automatic Speech Recognition (ASR)** model. This module is responsible for detecting and transcribing speech from the video's audio track. Depending on the specific requirements of the ASR model, an audio extraction sub-module may extract and pre-process the audio to ensure compatibility with the ASR module specifications. This step functions as an audio-to-text conversion. Subsequently, a **Machine Translation (MT)** module is invoked. Given the transcribed text, this component translates it from the source to the target language, completing the second stage, performing a text-to-text translation.

With the translated text obtained, the next stage involves **Speech Synthesis (SS)**. This stage utilizes a **Text-to-Speech (TTS)** sub-module, and optionally incorporates a **Voice Cloning (VC)** sub-module, depending on the TTS

sub-module's capabilities in voice conversion to maintain the original tone and style. The ASR, MT, and TTS modules collectively achieve Speech-to-Speech translation from the source to the target language.

To address the visual communication aspects, we employ a **Lip Synchronization (lip-sync)** model. Given an input audio track and a reference identity (e.g., a human face), this component generates a talking head where lip movements are synchronized with the translated speech. This finalizes our video translation pipeline, resulting in a translated video encompassing all facets of communication.

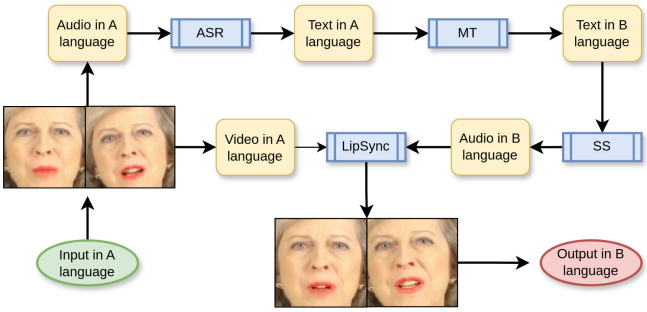

Fig. 1. Workflow of the Pipeline for Video Translation.

3.2 System Architecture

The proposed pipeline exhibits universal applicability across diverse domains, including video conferencing systems. However, its efficient utilization necessitates a rigorous system design due to inherent challenges.

Challenges. First, latency presents a significant concern in real-time environments such as video conferencing. This is particularly pertinent given that the pipeline incorporates multiple generative models processing various modalities: video, audio, and text. Each modality demands substantial computational resources, potentially disrupting the streaming process and compromising the real-time nature of the video conferencing experience, thereby impeding conversational fluidity. Second, the processing of these modalities requires considerable computational power, often necessitating high-performance GPUs to parallelize computations across models. This requirement is amplified by the pipeline's sequential nature, wherein each component's input depends on the preceding component's output. Consequently, standard parallel execution strategies such as round-robin are ineffective for enhancing performance and mitigating latency. Therefore, a meticulously designed execution strategy is crucial. Third, scalability in multi-user scenarios constitutes a critical challenge. Video conferencing typically involves multiple participants, each requiring real-time processing.

Scaling the system to accommodate numerous users exponentially increases computational demands, underscoring the need for optimized GPU utilization and scheduling strategies. While augmenting the number or power of GPUs can offer some mitigation, this approach is neither scalable nor cost-effective in the long term. Fourth, turn-taking and speaker management complexity pose a further challenge. Dynamic conversations are characterized by frequent speaker interruptions and transitions. The system must efficiently manage speaker changes while minimizing processing lag and ensuring seamless language switching. Fifth, personalization based on each participant's language preference is essential. A system hard-coded for specific source and target languages caters only to a limited subset of stakeholders, primarily those speaking the predetermined target language. In multilingual video conferencing, language agnosticism is a fundamental requirement.

To address these challenges, we designed a system architecture intended for integration with the proposed or similar pipelines to facilitate the development of a multilingual video conferencing system. This system aims to be cost-effective, modular, scalable, language-agnostic, and personalized, while maintaining uninterrupted real-time conversation. Figure 2 illustrates our proposed architecture.

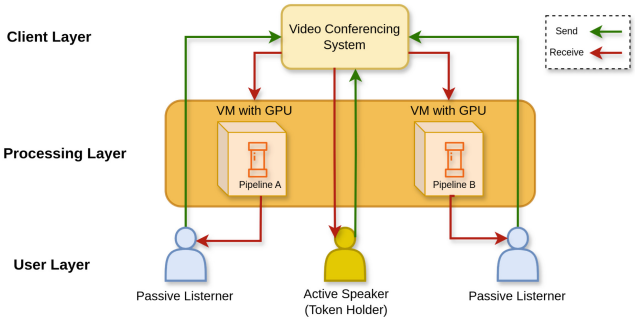

Fig. 2. Both the sending and receiving streams of the Active Speaker (in yellow) bypasses the processing layer, while the receiving stream of the Passive Listeners (in blue) passes through a GPU instance, which hosts the Video-translation pipeline that translates the video of the Active Speaker to target language (A or B) of the passive listeners. (Color figure online)

Layers. From a high-level perspective, the proposed architecture comprises three primary layers. The uppermost layer is the Video Conferencing System (Client Layer). This layer encompasses any video conferencing platform or interface, such as Jitsi, Teams, Google Meet, or Zoom, and manages user connections, video/audio transmission, and user interactions. The lowermost layer is the User Layer, where participants choose their preferred language for receiving translated video and interact directly with the conferencing client, mirroring the

user interaction model of conventional video conferencing systems. An intermediate Video-translation System (Processing Layer) is introduced to translate the video stream (encompassing both video frames and audio). This layer intercepts and processes media streams between users and the client before delivery to the end user.

Token Ring Mechanism. To clarify the operational principles, consider the following illustrative scenario. Initially, participants connect to the video meeting platform and interact with its user interface. Prior to initiating speech, based on meeting protocols or administrative decisions, one participant is assigned a 'Token' to speak, while others remain passive listeners. The token holder transmits and receives data via the media stream without intermediary processing. Conversely, passive participants select their desired target language for translation of the token holder's video. Each passive participant (listener) is allocated a dedicated GPU or a cloud-based GPU-equipped virtual machine from a pool of available GPUs. More specifically, each passive user is allocated a GPU instance upon selecting their target language. While listeners receive the video stream of the token holder, their incoming stream is routed through their allocated GPU instance, which hosts the pipeline. This pipeline translates the video into the listener's chosen target language before displaying it to the user. Upon completion of the speaker's turn, the token is passed to another participant, either by the current token holder or by request. The previous token holder, now a passive listener, is then allocated a GPU instance, and their incoming stream is subsequently processed by the video-translation pipeline, translating the video into their chosen language. The new token holder transmits and receives data without stream interception or intermediary processing. We term this turn-taking mechanism the 'Token Ring' strategy, as shown in Fig. 2.

The Token Ring strategy offers several advantages and effectively mitigates many of the aforementioned challenges. Crucially, by designating a single active speaker (token holder), each GPU assigned to a user is responsible for translation into only one target language. This approach addresses potential bottlenecks associated with concurrently operating multiple models within the pipeline in real-time environments. This single-target language focus is critical due to the pipeline's sequential and cascaded architecture, which precludes GPU parallelization of intra-pipeline model execution. Furthermore, without the Token Ring mechanism, if multiple speakers were active concurrently, system scalability would be compromised. A single GPU would necessitate running multiple pipeline instances, each with a different input language, introducing synchronization issues. The variable length of translated speech across languages would result in unordered delivery of translated videos from multiple speakers, disrupting conversational coherence. Our Token Ring turn-taking strategy not only resolves these challenges but also enhances scalability. To further quantify this, consider a video conferencing system without the Token Ring strategy with N participants. In a worst-case scenario where each participant speaks a different language, a single passive listener would require N-1 GPU instances. Scaling this

to an entire meeting with N participants, a total of $N \times (N-1)$ GPU instances would be required, resulting in a computational complexity of $O(N^2)$. However, with our Token Ring strategy, each allocated GPU is responsible for translation into only one target language, reducing the complexity to $O(N-1)$, which simplifies to $O(N)$. This approach is particularly advantageous when multiple passive listeners select the same target language. In this case, the number of allocated GPUs can be even smaller than the number of passive listeners, as multiple users will benefit from a single GPU instance that translates to the common target language, making the complexity $O(1) = \Omega(1)$. By decoupling the number of GPU instances from the number of participants, the Token Ring strategy ensures computational efficiency without sacrificing personalization. It guarantees that translated streams are delivered in sync with the speaker's turn, preserving conversational fluidity even in large meetings. This approach not only resolves the inherent challenges of latency, scalability, and dynamic speaker management but also provides a practical framework for deploying multilingual video conferencing systems in real-world settings. The combination of rigorous architectural design and strategic resource allocation makes our solution both robust and adaptable, addressing the demands of real-time communication.

Segmentation, Thresholding and Overlapping Buffering. Achieving low-latency, real-time communication in multimodal systems incorporating computationally intensive deep learning modules, such as generative lip-sync, presents a significant challenge. While optimization strategies targeting individual pipeline components and token-ring arbitration mechanisms offer partial mitigation, the inherent computational demands of these models remain a bottleneck to seamless user experience. Psycholinguistic studies establish a latency threshold of 100–300 ms as perceptually real-time, a benchmark commonly observed in video conferencing applications. However, the integration of supplementary pipelines with complex functionalities, particularly those involving generative image processing, necessitates innovative approaches to maintain comparable latency.

Our inference time analysis of the proposed pipeline, processing video chunks of varying duration (t), reveals a distinct relationship between chunk length and processing time (p). We observe a threshold (T) beyond which the processing time transitions to a logarithmic trend. This relationship is formally described by Eq. 1:

$$p(t) = \begin{cases} p > t : & t < T \\ \log t & : \ t \geq T \end{cases} \tag{1}$$

Prior to the threshold $T(t < T)$, the pipeline inference time p exceeds the video chunk duration t, indicating a system throughput less than unity. Conversely, for video chunk lengths surpassing $T(t \geq T)$, the inference time p becomes less than t, signifying improved throughput.

Based on this observation, we developed a segmented batched processing protocol to ensure continuous, near real-time conversational flow. This methodology introduces an initial latency equivalent to T. Subsequently, the active speaker's

video stream is segmented into fixed-duration chunks of length T. The initial pipeline iteration for the first segment incurs an inference time of $p = log(T)$. Upon completion of processing, the translated video segment is transmitted to the passive listener. While the listener consumes this T-second segment, subsequent video segments from the active speaker undergo concurrent processing within the pipeline.

As established by Eq. 1, for $t \geq T$, we have $p < t$, and specifically in our implementation with chunk size T, we have $p = log(T) < T$. This ensures that subsequent segments are processed in less time than the playback duration of the preceding segment. Consequently, despite the initial latency of T seconds introduced at the onset of the speaker's turn, continuous playback is achieved thereafter, providing a perceptually near real-time conversational experience characterized by uninterrupted translated video for the listener.

To sustain smooth playback, our segmented batched processing protocol leverages inverse throughput thresholding and overlapping buffering. We define reciprocal (inverse) throughput, $\tau(t)$, as the ratio of processing time to video chunk duration, as shown in Eq. 2:

$$\tau(t) = \frac{p(t)}{t} \tag{2}$$

The threshold T is determined as the minimum video chunk duration for which the inverse system throughput, $\tau(t)$, falls below 1.0 s. This condition, formalized in Eq. 3, ensures that the pipeline generates 1.0 s of translated video in less than 1.0 s processing time for $t \geq T$, thereby facilitating continuous playback after the initial latency. The condition for smooth playback and perceptually real-time communication is shown in Eq. 3:

$$\textbf{Condition: } \tau(t) < 1.0 \textbf{ for } t \geq T \tag{3}$$

By integrating token-ring turn-taking, input segmentation into fixed-duration chunks (T), inverse throughput thresholding to determine T, and overlapping buffering, the proposed system effectively addresses the challenges of latency in computationally intensive real-time multimodal communication. This integrated approach provides a robust and perceptually near real-time user experience.

4 Proof-of-Concept

In order to demonstrate the feasibility of implementing our proposed pipeline, we developed a simple yet comprehensive prototype that includes all the components outlined in the Pipeline subsection. This prototype was designed to run on modest and widely available computational units, such as GPUs, to ensure accessibility and scalability. By analyzing its performance and quality, we aim to provide a foundational framework for developers and businesses to build upon, enabling the development of next-generation video conferencing systems equipped with real-time multilingual video translation capabilities.

To simulate a real-world video conferencing environment, we implemented a simple user interface (UI) that mimics the experience of a multilingual video conferencing system. The prototype was intentionally kept as simple as possible, as our primary goal is to provide a general guide for developers and businesses. By maintaining simplicity, we ensure that the prototype is modular, scalable, cost-effective, and, most importantly, universal. By "universal," we mean that the system is both language-agnostic and platform-agnostic. Regardless of the selected models or video conferencing platform, developers can build upon this prototype to deploy a conferencing system equipped with video translation. This universality ensures that the system can be adapted to various use cases and environments, making it a versatile solution for real-world applications. As a result, simplicity is a key design principle throughout the development of this prototype.

4.1 Prototype

The prototype consists of four main modules: Automatic Speech Recognition (ASR), Machine Translation (MT), Text-to-Speech Synthesis (SS), and Lip Synchronization (LipSync). Each module was implemented using state-of-the-art open-source models to ensure high performance and adaptability. The ASR module utilizes the Whisper-Small model for speech recognition and language detection, chosen for its balance between speed and accuracy. The MT module employs the SeamlessM4T-Large model, which excels in producing accurate translations across multiple languages. For speech synthesis, the XTTS-v2 model was selected due to its ability to generate natural-sounding speech that closely mimics the original speaker's voice. Finally, the LipSync module uses the Wav2Lip-GAN model to synchronize the synthesized speech with the speaker's lip movements, ensuring a realistic viewing experience. The pipeline was implemented in a modular fashion, allowing each component to be tested and optimized independently before integration. This modular design not only facilitates performance tuning but also ensures scalability and flexibility, enabling future updates and enhancements. The entire system was deployed on Google Colab, leveraging its free-tier Nvidia T4 GPU capabilities for computational efficiency during the development and testing phases.

To simulate a real-world video conferencing environment, we developed a simple user interface (UI) using React.js, CSS, and HTML. The UI allows users to initiate video calls, select target languages for translation, and view the translated video with synchronized lip movements. Real-time communication between users was facilitated using Socket.io, while peer-to-peer video and audio streaming was implemented with the Simple-Peer library. The prototype was designed to be platform-agnostic, meaning it can be integrated into any existing video conferencing platform with minimal modifications. This universality ensures that the system can be adapted to various use cases and environments, regardless of the underlying technology stack.

4.2 Evaluation

The evaluation of the prototype was conducted using a combination of objective and subjective methods to assess both its technical performance and user experience. The objective evaluation focused on measuring the inference times of each module and the overall system throughput, which are critical for determining the system's suitability for real-time applications. The subjective evaluation, on the other hand, involved user surveys to gather qualitative feedback on the system's usability, translation quality, lip sync accuracy, and overall user experience.

For the objective evaluation, we designed an experimental setup using a dataset of 8-second-long videos in various languages. The videos were segmented into clips of 1, 2, 3, 5, and 8 s to test the system's performance under different input lengths. Each module's inference time was measured three times for each segment, and the average inference time along with the standard deviation was calculated. This approach allowed us to identify potential bottlenecks and assess the system's scalability.

The subjective evaluation involved ten participants who were asked to rate the translated videos based on several criteria, including Lip Sync Accuracy (LSA), Motion Naturalness (MN), Visual Quality (VIQ), and Vocal Quality (VOQ). Participants provided their feedback using a 5-point Likert scale, and the results were analyzed using the Mean Opinion Score (MOS) protocol.

5 Experiments and Results

Our experimental evaluation was designed to rigorously assess the proposed system across two critical dimensions: computational efficiency and user perception. These axes are paramount to validating the practicality and user-centricity of our multilingual video conferencing solution.

5.1 Objective Evaluation Results

The objective evaluation focused on quantifying the computational performance of our pipeline through precise measurements of module execution times and overall system throughput. To specifically examine the efficacy of our segmented batched processing protocol, described in Subsect. 3.2, we constructed a test dataset comprising 8-second video samples featuring speakers in various languages. These samples were meticulously segmented into clips of varying durations: 1, 2, 3, 5, and 8 s. This segmentation strategy was not arbitrary; it was deliberately chosen to empirically validate the predicted logarithmic-like trend in processing time relative to segment length, a core hypothesis underpinning our optimization strategy. By testing across these segment lengths, we aimed to identify the operational characteristics of our pipeline and confirm the benefits of segmented processing.

Each segment was then processed by the pipeline, instrumented with timers to capture module-specific inference times – for Automatic Speech Recognition

(ASR), Machine Translation (MT), Speech Synthesis (SS), and Lip Synchronization (LipSync) – alongside the total system processing duration. To ensure the robustness of our findings and account for potential variations in model inference times, each experiment was iterated three times, and standard deviations were calculated to assess the consistency of the recorded values. The primary metric for this objective assessment was inference time, directly reflecting the computational cost and real-time feasibility of each module and the integrated pipeline. For a representative analysis of Inference Time, we selected a German language video as input and targeted English as the translation output. This served as a benchmark scenario to illustrate the pipeline's typical performance profile. The detailed results of this analysis are presented in Table 1.

Table 1. Results of Inference Time Analysis (in seconds)

Length	ASR	MT	SS	LipSync	Total Inference	Reciprocal Throughput
1	0.2474	0.4280	0.6190	7.6944	8.9888	8.9888
2	0.2917	0.5524	1.3250	8.1047	10.2738	5.1369
3	0.3011	0.6691	1.4820	8.4715	10.9237	3.6412
5	0.3607	0.7651	1.9452	8.9386	12.0096	2.4019
8	0.4817	0.7648	2.1174	9.3346	12.6985	1.5873

The data presented in Table 1 robustly supports our hypothesis, detailed in Subsect. 3.2 and Eq. 1, that the total inference time p of the pipeline exhibits a non-linear, logarithmic-like relationship with the video chunk length t. This trend is clearly evidenced by the diminishing rate of increase in total inference time as video segment length increases. For example, while tripling the video length from 1 to 3 s, the total inference time only increases by approximately 22%, far below the 200% increase expected under linear scaling. Further extending the video length nearly threefold from 3 to 8 s results in a more modest increase of approximately 16% in inference time, far below the 166% increase expected under linear scaling. This sublinear scaling characteristic is precisely what our segmented batched processing protocol is designed to exploit to enhance efficiency within latency-constrained applications like video conferencing. The observed trend confirms the approximation $p \approx log(t)$, validating the core principle of our segmentation strategy for mitigating the computational burden of deep learning pipelines in real-time settings and fostering perceptually real-time communication with smooth playback.

It is important to note that, due to computational resource constraints inherent in utilizing a consumer-grade GPU (NVIDIA T4) – intentionally chosen to demonstrate accessibility and scalability on readily available hardware – our experiments did not achieve a Reciprocal Throughput (Inverse Throughput) value below the critical threshold of 1.0 s across the tested video lengths. Achieving $\tau(t) < 1.0$ for $t \geq T$, as stipulated in Eq. 3, represents the ideal smooth playback condition. Extrapolation suggests that smooth playback ($\tau < 1$) would occur at $t \approx 14$ on this hardware, demonstrating the protocol's scalability.

However, the observed trend of decreasing Reciprocal Throughput with increasing video length convincingly demonstrates the effectiveness of our fixed-length video segmentation and initial latency approach in substantially improving system efficiency and moving towards real-time performance, even within resource-constrained environments. This aligns with our prototyping philosophy of prioritizing open-source software and computationally economical processing units, underscoring the scalability and broad applicability of our proposed methodology, even when deployed on modest hardware configurations. Further optimization and deployment on higher-performance GPU infrastructure are anticipated to readily achieve the desired real-time throughput target.

To further assess the robustness of our pipeline performance, we conducted an analysis of the variability in inference times across repeated runs. Specifically, we examined the standard deviations of the inference times for each module across the three iterations performed for each video segment length. These standard deviation values, presented in Table 2, provide insights into the consistency and stability of the individual modules within the pipeline.

Table 2. Standard Deviations of Inference Times

Video Length	ASR	MT	SS	LipSync
1	0.0070	0.0013	0.1295	0.0972
2	0.0409	0.0130	0.2135	0.0873
3	0.0058	0.0145	0.1437	0.0586
5	0.0035	0.0065	0.1260	0.1195
8	0.0054	0.0044	0.2056	0.0827

The magnitude of standard deviation in inference times directly indicates module performance consistency. Notably, the Speech Synthesis (SS) module exhibited significantly higher standard deviations compared to ASR, MT, and LipSync, suggesting greater inference time variability and a potential need for system design considerations like buffering. Conversely, the consistently low standard deviations of ASR, MT, and LipSync indicate their stable and predictable performance across varying video lengths.

Furthermore, linguistic similarity between translated languages significantly influences pipeline performance, particularly in MT and SS modules. Linguistically closer languages may lead to faster processing, while distant languages can increase processing times and introduce variations in translated text length, potentially impacting output video duration and real-time synchronization. Future research should explore a wider range of language pairs to fully understand and address these language-dependent performance variations.

5.2 Subjective Evaluation Results

Subjective evaluation, crucial for gauging user perception and overall system usability, complemented our objective measurements. Recognizing the limitations of quantitative metrics in capturing nuanced aspects of user experience,

we employed a survey-based approach to gather qualitative feedback on key perceptual attributes of the translated videos. This evaluation aimed to assess the system's performance in real-world communication scenarios and to quantify user satisfaction across several critical dimensions. We defined the following criteria for subjective assessment:

The subjective evaluation employed four key criteria to assess user perception: Lip Sync Accuracy (LSA), gauging the temporal alignment of lip movements with the synthesized audio; Motion Naturalness (MN), evaluating the realistic and natural quality of lip movements beyond mere synchronicity; Visual Quality (VIQ), appraising the overall aesthetic fidelity of the generated video; and Vocal Quality (VOQ), assessing the synthesized voice's naturalness and similarity to the original speaker, especially considering voice cloning.

Survey participants were instructed to rate 8-second translated video segments within our video conferencing prototype environment across diverse language pairs using a 5-point Likert scale for each criterion. Collected responses were analyzed using the Mean Opinion Score (MOS) protocol to aggregate and interpret user feedback. The summarized subjective evaluation results, presented for each criterion and language pair, are detailed in Table 3. A more granular analysis, segmented by individual language pair, is presented below.

Table 3. Source - Target Translation Scores Based on Evaluation Criteria

Source - Target	LSA	MN	VIQ	VOQ	Overall
English – Turkish	3.2	3	3.2	3.4	3.2
Turkish – English	3.4	3.6	2.8	3.2	3.25
French – Spanish	4.6	4.8	4.8	4.8	4.75
Spanish – French	3.8	4.2	4.2	4.8	4.25
Arabic – German	4	4.4	2.8	4.6	3.95
German – Arabic	4.2	4.6	3.2	4.8	4.2

In summary, subjective evaluation revealed varying performance across language pairs. English-Turkish translations showed moderate user satisfaction with strong vocal quality but needing improvement in lip sync accuracy, motion naturalness, and visual quality. French-Spanish translations, particularly French-to-Spanish, demonstrated high user satisfaction across all metrics, likely due to linguistic similarity, with both directions achieving excellent vocal quality and lip sync accuracy. Arabic-German translations exhibited intermediate performance, with good motion naturalness and vocal quality, but consistently lower visual quality scores, indicating a clear area for future enhancement across these language pairs.

Overall, the subjective evaluation effectively highlighted performance variations across different language pairs, underscoring the influence of linguistic factors on user perception. Specifically, visual quality and, to a lesser extent,

lip synchronization accuracy emerged as key areas for targeted improvements in future iterations of the video translation pipeline, particularly for language pairs with greater linguistic divergence.

6 Conclusion

This paper introduced a novel, computationally efficient video translation method for multilingual video conferencing. Our pipeline, using open-source deep learning models, achieves perceptually real-time face-to-face translation. A "Token Ring" architecture and segmented batched processing protocol address latency and scalability. A simplified prototype demonstrated feasibility, with objective tests validating efficiency and subjective evaluations showing user satisfaction. Future work will optimize modules and improve visual quality and lip-sync, particularly for diverse language pairs. While a simplified prototype demonstrates the feasibility of our approach, future work will focus on rigorous evaluation with diverse datasets, exploring advanced lip sync models, and optimizing the system for large-scale deployments. We believe this work contributes a valuable framework for developing next-generation multilingual communication platforms.

Acknowledgments. This research paper has utilized Generative AI tools solely for the purpose of enhancing the readability and clarity of the manuscript. The AI was employed to refine sentence structure and improve grammatical flow, without altering the original content, introducing new ideas, or impacting the research findings. The authors retain full responsibility for the accuracy and integrity of the presented work. This research was made possible by the collaborative environment provided by **Aktif Bank**. The authors extend their appreciation for their support.

References

1. Aktas, M.S., Fox, G.C., Pierce, M.: Fault tolerant high performance information services for dynamic collections of grid and web services. Futur. Gener. Comput. Syst. **23**(3), 317–337 (2007)
2. Aydin, G., Aktas, M.S., Sayar, A.: Servogrid complexity computational environments CCE integrated performance analysis. In: 2005 6th International Workshop on Grid Computing (GRID), pp. 256–261 (2005)
3. Bao, H., et al.: MILG: realistic lip-sync video generation with audio-modulated image inpainting. Vis. Inform. **8**(3), 71–81 (2024)
4. Casanova, E., et al.: XTTS: a massively multilingual zero-shot text-to-speech model. arXiv preprint arXiv:2406.04904 (2024)
5. Cheng, X., et al.: Transface: unit-based audio-visual speech synthesizer for talking head translation. arXiv preprint arXiv:2312.15197 (2023)
6. Chu, S.N., Goodell, A.J.: Synthetic patients: simulating difficult conversations with multimodal generative AI for medical education. arXiv preprint arXiv:2405.19941 (2024)

7. Das, D., Biswas, S., Sinha, S., Bhowmick, B.: Speech-driven facial animation using cascaded GANs for learning of motion and texture. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020, Proceedings, Part XXX 16, pp. 408–424. Springer (2020)
8. Hong, F.T., Zhang, L., Shen, L., Xu, D.: Depth-aware generative adversarial network for talking head video generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3397–3406 (2022)
9. Kheddar, H., Hemis, M., Himeur, Y.: Automatic speech recognition using advanced deep learning approaches: a survey. Inf. Fusion 102422 (2024)
10. Kim, M., Hong, J., Ro, Y.M.: Lip to speech synthesis with visual context attentional GAN. Adv. Neural. Inf. Process. Syst. **34**, 2758–2770 (2021)
11. KR, P., Mukhopadhyay, R., Philip, J., Jha, A., Namboodiri, V., Jawahar, C.: Towards automatic face-to-face translation. In: Proceedings of the 27th ACM International Conference on Multimedia, pp. 1428–1436 (2019)
12. Li, C., et al.: Latentsync: audio conditioned latent diffusion models for lip sync. arXiv preprint arXiv:2412.09262 (2024)
13. Li, R., Pu, D., Huang, M., Huang, B.: Unet-TTS: improving unseen speaker and style transfer in one-shot voice cloning. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8327–8331. IEEE (2022)
14. Lin, G., Jiang, J., Yang, J., Zheng, Z., Liang, C.: Omnihuman-1: rethinking the scaling-up of one-stage conditioned human animation models. arXiv preprint arXiv:2502.01061 (2025)
15. Malik, M., Malik, M.K., Mehmood, K., Makhdoom, I.: Automatic speech recognition: a survey. Multimedia Tools Appl. **80**, 9411–9457 (2021)
16. Nacar, M.A., Aktas, M.S., Yuen, D.A.: Vlab: collaborative grid services and portals to support computational material science. Concurr. Comput. Pract. Exp. **19**(12), 1717–1728 (2007)
17. Olmezogullari, E., Aktas, M.S.: Representation of click-stream data sequences for learning user navigational behavior by using embeddings. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 3173–3179 (2020)
18. Olmezogullari, E., Aktas, M.S.: Pattern2vec: representation of clickstream data sequences for learning user navigational behavior. Concurr. Comput. Pract. Exp. **34**(9) (2022)
19. Oskooei, A.R., Babacan, M.S., Yağcı, E., Alptekin, Ç., Buğday, A.: Beyond synthetic benchmarks: assessing recent LLMs for code generation. In: 14th International Workshop on Computer Science and Engineering, WCSE (2024)
20. Prabhavalkar, R., Hori, T., Sainath, T.N., Schlüter, R., Watanabe, S.: End-to-end speech recognition: a survey. IEEE/ACM Trans. Audio Speech Lang. Process. (2023)
21. Prajwal, K., Mukhopadhyay, R., Namboodiri, V.P., Jawahar, C.: A lip sync expert is all you need for speech to lip generation in the wild. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 484–492 (2020)
22. Rafiei Oskooei, A., Aktaş, M.S., Keleş, M.: Seeing the sound: multilingual lip sync for real-time face-to-face translation. Computers **14**(1), 7 (2024)
23. Rafiei Oskooei, A., Yahsi, E., Sungur, M., S. Aktas, M.: Can one model fit all? An exploration of wav2lip's lip-syncing generalizability across culturally distinct languages. In: International Conference on Computational Science and Its Applications, pp. 149–164. Springer (2024)

24. Ritter, M., Meier, U., Yang, J., Waibel, A.: Face translation: a multimodal translation agent. In: AVSP'99-International Conference on Auditory-Visual Speech Processing. Citeseer (1999)
25. Sahinoglu, M., Incki, K., Aktas, M.S.: Mobile application verification: a systematic mapping study. In: Computational Science and Its Applications–ICCSA 2015. Lecture Notes in Computer Science, vol. 9159, pp. 147–163 (2015)
26. Shen, S., et al.: Difftalk: crafting diffusion models for generalized audio-driven portraits animation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1982–1991 (2023)
27. Shin, A.H., Lee, J.H., Hwang, J., Kim, Y., Park, G.M.: Wav2nerf: audio-driven realistic talking head generation via wavelet-based nerf. Image Vis. Comput. 105104 (2024)
28. Song, H.K., et al.: Talking face generation with multilingual TTS. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 21425–21430 (2022)
29. Stypułkowski, M., Vougioukas, K., He, S., Zięba, M., Petridis, S., Pantic, M.: Diffused heads: diffusion models beat GANs on talking-face generation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 5091–5100 (2024)
30. Sun, X., et al.: Vividtalk: one-shot audio-driven talking head generation based on 3D hybrid prior. arXiv preprint arXiv:2312.01841 (2023)
31. Team, N., et al.: Scaling neural machine translation to 200 languages. Nature **630**(8018), 841 (2024)
32. Tufek, A., Gurbuz, A., Aktas, M.S.: Provenance collection platform for the weather research and forecasting model. In: 2018 14th International Conference on Semantics, Knowledge and Grids (SKG), pp. 17–24 (2018)
33. Vougioukas, K., Petridis, S., Pantic, M.: End-to-end speech-driven realistic facial animation with temporal GANs. In: CVPR Workshops, vol. 887, pp. 37–40 (2019)
34. Wang, H., Wu, H., He, Z., Huang, L., Church, K.W.: Progress in machine translation. Engineering **18**, 143–153 (2022)
35. Xu, S., et al.: Vasa-1: lifelike audio-driven talking faces generated in real time. arXiv preprint arXiv:2404.10667 (2024)
36. Ye, Z., et al.: Geneface++: generalized and stable real-time audio-driven 3D talking face generation. arXiv preprint arXiv:2305.00787 (2023)
37. Ye, Z., et al.: Mimictalk: mimicking a personalized and expressive 3D talking face in minutes. arXiv preprint arXiv:2410.06734 (2024)
38. Ye, Z., et al.: Real3d-portrait: one-shot realistic 3D talking portrait synthesis. arXiv preprint arXiv:2401.08503 (2024)
39. Yin, F., et al.: Styleheat: one-shot high-resolution editable talking face generation via pre-trained stylegan. In: European Conference on Computer Vision, pp. 85–101. Springer (2022)
40. Zhu, W., et al.: Multilingual machine translation with large language models: empirical results and analysis. arXiv preprint arXiv:2304.04675 (2023)