Temporally Streaming Audio-Visual Synchronization for Real-World Videos

Jordan Voas* Wei-Cheng Tseng* Layne Berry Xixi Hu Puyuan Peng
James Stuedemann David Harwath
The University of Texas at Austin
{jvoas, raytseng, layne.berry, hxixi, pyp, jbstuedemann, harwath}@utexas.edu

Abstract

We introduce RealSync, a novel dataset designed to significantly enhance the training and evaluation of models for audio-visual synchronization (AV Sync) tasks. Sourced from high-quality YouTube channels, RealSync covers a wide range of content domains, providing an improved scale, diversity, and alignment with broadcast content compared to existing datasets. It features extended-length video samples, catering to the critical need for more comprehensive, real-world training and evaluation materials. Alongside this dataset, we present StreamSync, a model tailored for real-world AV Sync applications. StreamSync is designed to be backbone agnostic and incorporates a streaming mechanism that processes consecutive video segments dynamically, iteratively refining synchronization predictions. This innovative approach enables StreamSync to outperform existing models, offering superior synchronization accuracy with minimal computational cost per iteration. Together, our dataset and the StreamSync model establish a new benchmark for AVSync research, promising to drive the development of more robust and practical AVSync methods. https://github.com/jvoas655/StreamSync

1. Introduction

Multimedia experiences critically depend on the seamless integration of audio and visual streams. However, these streams, though often captured concurrently, are typically encoded and streamed separately to target devices. This separation can introduce synchronization errors due to variations in encoding processes and transmission delays, significantly degrading the playback quality and potentially causing communication misunderstandings in scenarios such as video conferencing and live broadcasts. The increasing reliance on digital streaming across diverse content types—from television broadcasts to live streaming—underscores the importance of robust, automatic detection of these synchronization discrepancies.

Current advancements in audio-visual synchronization (AVSync) primarily focus on aligning audio cues with visual elements like lip movements to support related tasks such as lip-reading and active speaker detection [3, 8, 9, 24, 27]. However, these techniques are often tailored to specific scenarios and fail to address the broader challenge of AV Sync across varied, real-world conditions. Additionally, they do not account for resource constraints typical of many target devices, which lack the computational power to handle complex synchronization checks. This is a significant issue even in high-resource environments such as sporting events, where multiple camera feeds preclude allocating high-compute resources like GPUs to continuously monitor each stream for synchronization errors.

To address these challenges, we introduce RealSync, a comprehensive dataset of real-world broadcast videos from high-quality sources. RealSync is specifically curated for professional broadcast environments, filling a gap by providing high-quality, aligned content. It encompasses a broad array of real-world synchronization scenarios, including diverse settings in news and sports which introduce distinct audio-visual contexts. Compared to existing AVSync datasets, RealSync provides significant advantages. Single-domain datasets like LRS3 [7] lack suitable diversity, while open-domain datasets like VGGSound [4], though diverse, often suffer from misalignment issues with professional content. RealSync also greatly extends clip durations to support novel task formulations and more realistic evaluations of synchronization patterns, while detailed audio event annotations enable more in-depth performance analysis.

Building on these findings, we propose StreamSync, a novel AV Sync method that optimizes the SparseSync architecture [16]. StreamSync is engineered to leverage the persistent nature of real-world synchronization errors and is particularly suited for environments with limited computational resources. It enhances detection capabilities by accumulating predictive information from one synchronization check to the next, significantly outperforming existing methods with minimal computational overhead.

[*] Equal Contribution First Author.

Our primary contributions are threefold.

1. We present a diverse, high-quality dataset that enhances AV Sync model robustness by encompassing a broad range of real-world broadcast synchronization scenarios.

2. We highlight key challenges for AVSync, particularly the limitations of current models under resource constraints and the inadequacies of existing datasets in accurately representing real-world content features.

3. We introduce StreamSync, a novel method that efficiently improves general AV Sync model performance, exploiting persistent synchronization errors for improved detection with minimal computational demands, targeted for real world resource constrained environments.

2. Related Works

The advent of streaming applications underscores the necessity for advanced audio-visual synchronization (AV Sync) techniques. Initial research efforts in this domain have predominantly focused on lip-syncing or speech synchronization, where speech signals are aligned with corresponding lip movements [13, 18, 20, 26]. This strong correlation between auditory and visual cues serves as a robust foundation for model training and has proven effective in extending to tasks like lip-reading [27], lip-shape generation [25], and active speaker detection [15, 27]. The integration of attention mechanisms represents a significant breakthrough, enhancing the efficacy of AV synchronization technologies [19]. Modern studies have adopted sophisticated architectures, such as the dual-encoder system in SyncNet [8] and cross-modal contrastive learning in ModEFormer [12], to boost performance. Furthermore, the PerfectMatch [9] approach treats synchronization as a cross-modal retrieval challenge, seeking to accurately match audio-visual pairs from a selection of candidates.

Concurrently, there has been a focus on synchronizing video content with corresponding audio cues, especially challenging due to the fleeting nature of sound events and the intricate link between visual objects and their sounds. Initiatives like AVE-Net [2] aim to synchronize the sounds of musical instruments with their visual representations. In the realm of sports, TennisED [10] utilizes distinct audio and video event detectors to synchronize tennis match streams at the moments the ball is struck. Other research has explored synchronizing through human motion analysis [22] and the alignment of commercial breaks [23].

However, these methods often cater to niche applications and exhibit limited generalizability. For example, SyncNet struggles with scenarios involving voice-overs or rapid scene changes [23], common in broadcast environments, while TennisED excels primarily in scenarios marked by distinct, high-impact sounds, such as a tennis ball being hit [10]. In response, recent efforts have sought to develop more adaptable AV Sync models. Chen et al. [3] introduced a transformative approach utilizing Transformers [29] to model spatial-temporal relationships across modalities, facilitating sound source localization and broadening the scope of AV Sync applications. Similarly, Iashin et al. [16] proposed SparseSync, a multi-modal architecture adept at identifying synchronization cues in 'sparse' video events, setting new benchmarks for synchronization detection. Recently, Iashin et al. [17] used contrastive pretraining and increased model scale to improve synchronization capability.

Despite these advancements, the prevailing datasets used for AV Sync research are often too narrow in scope and fail to encompass the diversity and complexity of real-world situations, mainly consisting of videos from specialized domains or focused on user-generated content rather than production grade content [6, 7, 10]. Such limitations inhibit models' ability to generalize across varied contexts and impede the development of robust, versatile AV Sync models, particularly as the brief duration of video clips in these datasets does not adequately mimic the synchronization challenges in streaming applications [4, 11]. Our work addresses these gaps with a novel dataset designed to capture the diverse and complex nature of real-world settings, paving the way for more effective and applicable solutions.

3. RealSync Dataset

To establish a benchmark for audio-visual synchronization tasks, we selected high-quality YouTube channels with content akin to television broadcasts and live streaming. Our dataset encompasses nine channels: five sports (golf, football, ice hockey, baseball, and tennis) and four news (CBS, ABC, NBC, and CNN). We ensured a balanced collection by gathering a similar duration of videos from each channel, totaling 670 videos and 927 hours. Each video adheres to strict criteria: a minimum length of 20 minutes, resolution of at least 720p, and standard bitrate.

Our video preprocessing pipeline standardizes the format to enhance dataset utility. All videos were encoded using the H.264 codec at 25 fps for video and AAC codec for audio, ensuring single-channel, 16-bit, 16kHz quality[1]. To mitigate I/O bottlenecks, videos were segmented into 5-minute clips, improving processing and loading efficiency. Videos were also downsampled to a minimum of 256 pixels on the shortest dimension, maintaining aspect ratios to balance visual detail with computational efficiency.

3.1. Evaluation and Data Annotation

Effective benchmarks necessitate uniform evaluation and thorough annotation for interpreting results. We partitioned our dataset into 90% for training, and 5% each for development and testing sets. For uniform evaluation, random-

[1] Video and audio encoding were executed using `ffmpeg`, employing H.264's High Profile and AAC's Low Complexity profile.

length segments were sampled from each clip in the evaluation sets, producing about 32,000 segments. Fixed offsets for each segment were sampled from a truncated normal distribution between ±2 seconds. Adhering to ITU-T J.248 guidelines [1], which suggest a maximum acceptable delay of 225 ms for human perception, we divided the ±2 seconds interval into 21 offset bins, each covering 0.2 seconds, simplifying the regression task into a classification task. We also collect the following per segment annotations for specific synchronization sub-classes to facilitate precise evaluation of the failure and success cases for future models.

Identifying Talking Heads: Video segments with talking heads are particularly telling for evaluating audio-visual synchronization, as the visible speech movements provide clear synchronization cues. We developed a dedicated pipeline to detect talking heads, described in Section A.1 of the supplementary material. This process involves detecting faces in video frames and analyzing these frames with their corresponding audio using a pre-trained lip-syncing model [8]. Segments are categorized as 'talking head' if the detected face corresponds with the audible speech, providing direct feedback on synchronization accuracy. Conversely, segments where the speech does not match the visible individual are marked as 'voice-over,' representing a more challenging scenario for synchronization models that primarily rely on visual speech cues.

Recognizing Audio Events: The presence and type of audio events significantly influence the performance of AV Sync models. Clear audio cues, such as the sound of a ball hitting a racket, facilitate precise synchronization, whereas ambiguous sounds like applause or background noise pose greater challenges. We annotated each video clip with identified audio events to better understand model performance across different sound contexts. The detailed process for audio event identification is outlined in Section A.2, involving segmenting the audio track into overlapping chunks and classifying these using an off-the-shelf audio classifier [5]. This annotation supports the development of models capable of handling diverse audio-visual scenarios.

Dataset	$\mathbf{N_{clips}}$	$\bar{\mathbf{T}}_{\mathbf{clips}}$	$\mathbf{T_{total}}$	**Domain**	**AVC**
AudioSet [11]	2.1m	10s	243d	General	✗
AVE [28]	4.1k	10s	11.5h	General	✓
TennisED [10]	4	1.5h	6h	Sports	Δ
VGGSound [4]	200k	10s	550h	General	Δ
VGGSSync [3]	100k	10s	275h	General	✓
LRS2 [6]	118k	6.8s	224h	News	✓
LRS3 [7]	74.5k	22.9s	474h	TED Talks	✓
RealSync	11.2k	5m	927h	Sports/News	Δ

Table 1. Statistics for common datasets in AV Sync. $\mathbf{N_{clips}}$ is the total number of clips within the dataset; $\bar{\mathbf{T}}\mathbf{clips}$ is the average duration of each clip; $\mathbf{Ttotal}$ is the aggregate duration of the dataset; **AVC** indicates whether the audio and video components correspond. A Δ symbol indicates that the sound source is visually discernible in the video, though synchronization is not assured.

Channel	$\mathbf{N_{vid}}$	$\mathbf{N_{clip}}$	$\mathbf{T_{to}}$	$\mathbf{T_{TH}}$	$\mathbf{T_{VO}}$
ABCNews	40	1275	106.25	37.16	18.21
CBSNews	33	1208	100.67	45.48	22.81
CNN	93	1277	106.41	57.31	28.24
NBCNews	174	1451	120.92	41.63	26.83
GolfsHome	51	1026	85.50	35.14	24.15
MLB	59	1348	112.33	6.35	8.79
NFL	49	1185	98.75	6.45	7.60
NHL	141	1181	98.42	10.16	3.91
WTA	30	1173	97.75	4.31	9.41
Total	670	11124	927.00	244.00	149.96

Table 2. Channel-wise statistics for the proposed dataset. $\mathbf{N_{vid}}$: number of raw videos; $\mathbf{N_{clip}}$: number of clips; $\mathbf{T_{to}}$: total duration (hours) of videos; $\mathbf{T_{TH}}$: total duration (hours) of talking-head segments; $\mathbf{T_{VO}}$: total duration (hours) of voice-over segments.

3.2. Statistics

This section provides detailed statistics of the RealSync dataset, as illustrated in Tab. 1. Comprising 11,124 five-minute clips from 670 videos, our dataset offers an unprecedented scale and diversity, particularly valuable in sports and news contexts where synchronization discrepancies are prevalent. The extended duration of clips facilitates the exploration of methods capable of modeling longer contexts, essential for practical applications. While our dataset strives for precise audio-video alignment, we acknowledge the potential for unsynchronizable instances occurring for some segments, reflecting real-world challenges.

We provide comprehensive channel-specific statistics in Tab. 2, showing a balanced collection across channels, each contributing close to a hundred hours of video. The distribution of talking-head annotations, detailed in Sec. 3.1, illustrates the distinct broadcasting styles of different channels. For instance, sports channels often feature less frequent talking-head segments than news channels, which regularly show news anchors and reporters. Fig. 1 presents the distribution of audio events within the dataset, highlighting the predominance of speech-related events alongside a variety of other sound types. This diverse range of audio cues enhances the dataset's utility for training models to recognize and synchronize a wide array of audio-visual inputs, reflecting the complexity of real-world AV Sync tasks.

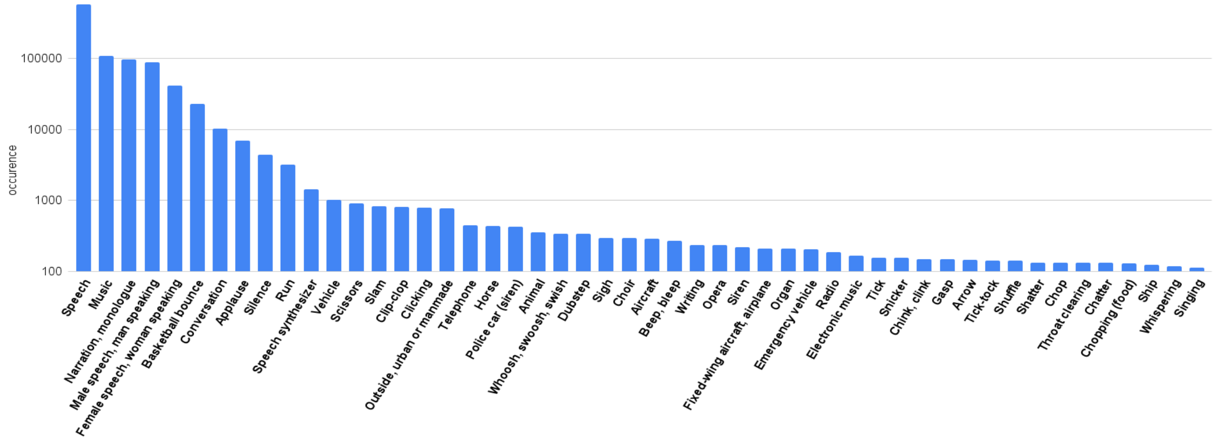

Figure 1. Occurrences of each audio event within RealSync. Due to space limitations, only the top fifty audio events are shown here.

4. StreamSync

Traditional approaches to AV Sync typically focus on short video segments to manage computational constraints effectively. For instance, SparseSync [16] employs five-second windows for each inference step, shaped by two primary challenges. The first is the scarcity of training data suitable for modeling AV Sync over extended durations. Most datasets, such as VGG-Sound [4], confine their samples to no more than 10 seconds, limiting model applicability to longer, real-world content. Our dataset, RealSync, overcomes this limitation by including samples up to five minutes, enabling more realistic AV Sync development.

The second challenge relates to computational efficiency. Modern transformer architectures cause AV Sync computational costs to rise quadratically with the increase in input context size. This growth is impractical for AV Sync tasks requiring low latency on devices with limited computational resources. Extending the analysis to longer video sequences could theoretically improve performance, especially in contexts with sparse synchronizable events. However, such an extension would compromise system practicality. For example, the Synchformer [17], despite outperforming SparseSync, requires nearly ten times the resources for a single prediction (Tab. 3) on a high-end GPU and may scale even less favorably on typical consumer devices.

Given these constraints, enhancing the performance of AV Sync models without raising per-prediction costs is crucial. An analysis of prior AV Sync models revealed that a significant source of inaccurate predictions could be attributed to seemingly random prediction errors, which are uniformly distributed and not strongly correlated even when predictions are made on closely sequenced video windows. To illustrate this, we analyzed the performance of an SparseSync model fine-tuned on our RealSync dataset with offset predictions collected with only a small temporal window shift. Our findings, detailed in Fig. 2, show that significant errors (greater than 0.2 seconds from the correct offset) are mostly uniformly distributed across different classes. Moreover, over 50% of the errors display a random normal distribution across consecutive, largely overlapping, windows, indicating these errors are uncorrelated.

These insights prompted the development of Accumulated Probability Averaging (APA), which significantly boosts SparseSync's performance by averaging probabilities over multiple video windows. APA is computationally efficient and well-suited for real-world applications that require continuous monitoring of synchronization errors. Furthermore, our evaluations of APA prompted an investigation into whether a model trained explicitly for this accumulated window technique could surpass even APA methods. Such a model could not only mitigate errors but also provide deeper insights into prediction trends by effectively expanding its context window, thus overcoming the inherent limitations of short-context synchronization. Further, while APA would inherently introduce a delayed response to synchronization changes, due to their averaging nature, learned methods could potentially adapt faster to such issues.

Building on these developments, we introduce StreamSync, a novel framework that redefines the AV Sync task through recurrent snapshot predictions. This architecture enhances the SparseSync model by significantly increasing the considered context size without impacting computational efficiency or execution latency. StreamSync is designed to be dynamically scalable and maintains performance on par with baseline models, making it a robust solution for diverse AV synchronization scenarios. Further, while not investigated in this work, a significant limitation of APA would be an inherent delayed response to changing synchronizations. A learned framework such as StreamSync could be expected to be better capable of responding to sudden changes in synchronization if trained for such.

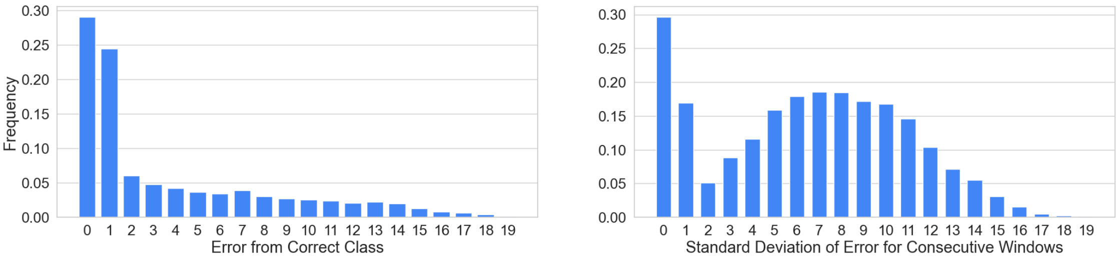

Figure 2. Analysis of Prediction Error and Standard Deviation Across Consecutive Video Windows. On the left, the distribution of prediction error from the correct label highlights mostly uniform error distribution when the predicted class is significantly incorrect. On the right, the distribution of standard deviations for these errors across consecutive windows reveals a random normal error distribution, underscoring the lack of correlation between errors in closely sequenced video windows.

The architecture of the StreamSync model, as illustrated in Fig. 3, enhances the functionality of a backbone Snapshot Predictor, based on SparseSync [16] in our evaluations, by generating snapshot predictions of the AV offset and forwarding a learned Streaming Token. Initially, Sparse Selectors identify pivotal frames from both audio and visual modalities. These frames, along with learned classification (*CLS*) and modality separator (*MOD*) tokens, are processed by a Transformer Encoder to classify offsets. A unique feature of StreamSync is the introduction of the passthrough (*PASS*) token. While it does not directly influence snapshot predictions, it facilitates communication of the model's confidence in its current predictions. The *CLS* token undergoes processing through a linear layer to determine offset labels, and together with the *PASS* token, it predicts a streaming token S_i for each video window i. This sequence, governed by a predetermined window hop size that aligns with the frequency of synchronization checks, accumulates a sequence of Streaming Tokens $< S_i : S_{i+H} >$.

At each synchronization snapshot, StreamSync calculates the snapshot offset prediction and utilizes the history of Streaming Tokens to compute a streaming prediction. This mechanism incorporates the Streaming Tokens sequence with sinusoidal positional embeddings and an additional *CLS* token, processed by another Transformer Encoder (Streaming Head). The resultant output from this *CLS* token is then mapped through a linear layer to produce a Streaming Prediction of the AV offset. Notably, our framework is backbone agnostic, and could apply to alternative pretrained Snapshot Predictors, since they only require the ability to insert or adapt a output feature to produce a Streaming Token, allowing future advancements in snapshot AV Sync to easily propagate to the Streaming setting as well for greater efficiency or accuracy.

4.1. Training

The training process of StreamSync involves introducing artificial audio stream offsets within a range of ±2 seconds, discretized into 21 distinct class labels with 0.2-second intervals. To counter potential overconfidence in predictions, label smoothing is applied with a smoothing factor of 0.1. The loss (Eq. (1)) function integrates cross-entropy calculations for both snapshot and streaming predictions, catering to the model's dual predictive capabilities: immediate snapshot accuracy and sustained streaming accuracy.

$$\mathcal{L} = -\frac{1}{H}\sum_{i=1}^{H}\left[\alpha\sum_{c=1}^{C} y_c^i \log(p_{o,c}^i) + \beta\sum_{c=1}^{C} y_c^i \log(p_{s,c}^i)\right] \tag{1}$$

Here, y_c^i denotes the actual class label for class c at iteration i, while $p_{o,c}^i$ and $p_{s,c}^i$ represent the predicted probabilities for snapshot and streaming predictions, respectively. α and β are hyperparameters used to balance the emphasis between snapshot and streaming prediction accuracies.

Computation of Eq. (1) entails generating Streaming Tokens over H iterations per sample, markedly increasing memory requirements due to gradient backpropagation over all H tokens. A two-phase training strategy mitigates this: initially, the model employs pre-trained SparseSync weights, freezing all but the new components (Streaming Head, Projection Layers, and *PASS* token) to efficiently propagate gradients through all H iterations simultaneously. Then in phase 2 all weights are unfrozen, improving finetuning capability and the ability to shape temporal patterns in Streaming Tokens, but backpropagation is limited to K randomly selected consecutive iterations out of H, balancing parameter efficiency with temporal considerations.

Short experiments were utilized to establish optimal hyperparameters, as follows: history length $H = 12$, a phase 2 backpropagation cap of $K = 3$, and a one-second window stride. Parameters $\alpha = 0.5$ and $\beta = 1.0$ are set, emphasizing streaming prediction accuracy. Phase 1 training runs for 500 iterations, followed by 3 epochs in phase 2. We examine Streaming Heads with both a three-layer, 512-dimensional Transformer (StreamSync-Trf) and a three-layer, 512-dimensional LSTM alternative (StreamSync-LSTM). The training was done with batch

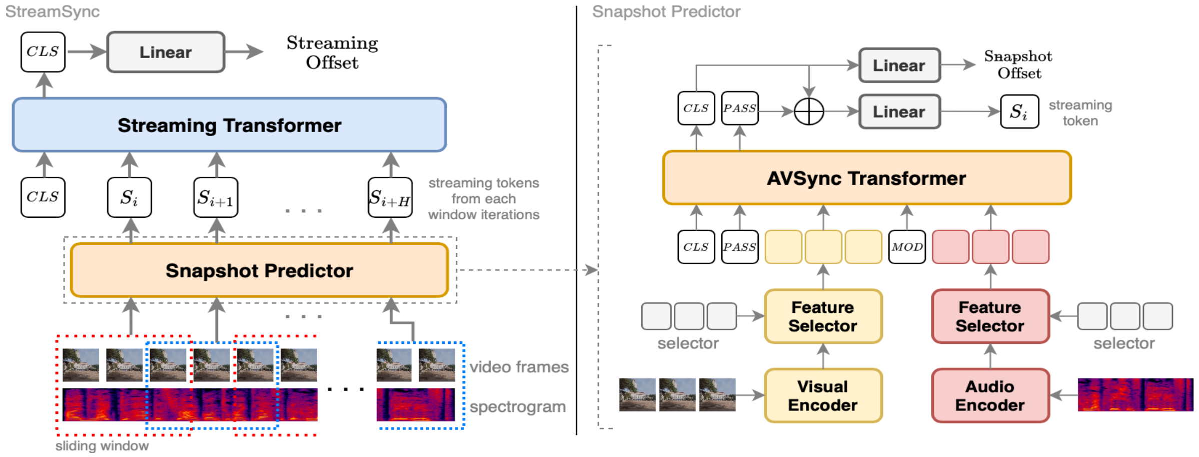

Figure 3. The architecture of the proposed StreamSync model. **Right**: Modified SparseSync architecture for snapshot predictions and generation of Streaming Tokens **Left**) StreamSync architecture integrating sliding window samples of inputs through Snapshot Predictor and Streaming Transformer. This figure is for illustrative purposes only. In real-world applications and our evaluations, input windows are processed sequentially, not in batch as illustrated. AV offset is held consistent over all Streaming Tokens.

size 16 and a learning rate of $4e-5$ with linear decay. Hyperparameters ablations are documented in Sec. 5.2.

5. Results

We begin by summarizing key results from evaluating our StreamSync model against various baselines, including a non-finetuned SparseSync, a finetuned SparseSync on the RealSync dataset, the same finetuned SparseSync with APA, a non-finetuned Synchformer with and without APA, and our novel StreamSync model. All models underwent testing under identical conditions, employing a 2-second sample window hop size—identified as optimal compared to the 1-second interval used during training, as outlined in Sec. 5.2. Each model was assessed using 18 streaming frames, with StreamSync and APA metrics reported at the final (18th) iteration. For fairness, metrics for both the finetuned and non-finetuned SparseSync models without APA, and Synchformer, are reported at the best prediction observed across all iterations, negating any advantages from increased input in the streaming models.

Model evaluation employed exact match accuracy and ±1 class tolerance accuracy. For a thorough analysis, we also utilized Mean Average Precision (mAP) [21] and Receiver Operating Characteristic Area Under the Curve ($R_{oc}A_{uc}$) [14], providing deeper insights into model performance. Additional ablations and detailed examinations of StreamSync's effectiveness, especially across specific event types within the RealSync dataset, were conducted.

mAP: Evaluates the model's precision in predicting across different classes and their confidence-ranked ordering. It offers a nuanced perspective on the model's capability to not only predict accurately but also to rank these predictions effectively. mAP is particularly beneficial in scenarios demanding fine distinctions among similar classes.

	Acc	Acc^{tol1}	$R_{OC}A_{UC}$	mAP	ms
Pt SparseSync*	0.207	0.381	0.717	0.177	18.5
Ft SparseSync*	0.381	0.598	0.872	0.430	18.5
Pt Synchformer*	0.369	0.594	0.865	0.410	171
APA SparseSync	**0.580**	**0.847**	0.953	0.585	18.4
StreamSync-Trf	0.566	0.841	0.959	0.621	20.6
StreamSync-LSTM	0.567	0.845	**0.961**	**0.632**	19.2
APA Synchformer	0.520	0.795	0.938	0.564	171

Table 3. Main results for all novel and baseline models tested. All models were tested over 18 identical streaming iterations. Models with * are not streaming capable and so use the best prediction among the iterations. APA SparseSync is finetuned while APA Synchformer is not. Inference cost evaluations were ran over 100 samples on an otherwise idle system, measured on the 18th iteration. A single L4 24GB GPU was utilized with a batch size of 1.

ROCAUC: Measures the model's ability to differentiate between classes at various thresholds, indicative of its confidence in class identification. A high ROCAUC score (max of 1.0) signifies the model's proficiency in discerning subtle differences between classes, crucial for precise classification in complex situations.

5.1. Evaluation Results

Tab. 3 shows the StreamSync model's effectiveness in tackling a broad range of synchronization challenges in our dataset. The pretrained SparseSync, with an accuracy of 20.7% and a mAP of 17.7%, reveals significant limitations

Window Counts	2	4	12	24	32	40	48
Ft SparseSync	0.48	0.51	0.56	0.59	0.61	0.60	0.62
StreamSync-Trf	0.46	0.52	0.60	0.64	0.65	0.67	0.68

Table 4. mAP results for extended window counts.

in addressing the diverse synchronization issues found in real-world content. This performance discrepancy highlights the inadequacies of traditional datasets for realistic synchronization tasks. However, finetuning SparseSync on RealSync significantly boosts its performance, increasing accuracy to 38.1% and mAP to 43.0%. This marked improvement, though anticipated due to distributional discrepancies, accentuates the content distribution gap between RealSync and prior alternatives, underscoring the advantages of training with data that mirrors broadcast content.

Implementing APA with SparseSync across 18 streaming windows further enhances its capabilities, elevating accuracy to 58.0% and mAP to 58.5%. This approach, which aggregates predictions over time, significantly enhances robustness and excels in accuracy metrics. StreamSync, using 18 windows of streaming history, not only matches APA SparseSync in accuracy but also exceeds it in critical metrics like ROCAUC and mAP, reaching scores of 0.959 and 62.1%, respectively. These results indicate that StreamSync may more effectively capitalizes on temporal relationships over extended periods, leveraging high-level features for superior performance in predicting synchronization offsets.

Furthermore, we show that Synchformer, even unfinetuned, nearly matches the performance of finetuned SparseSync. When combined with APA, Synchformer's performance boost is akin to that of APA SparseSync, suggesting similar distributional challenges that streaming methods alleviate. Although Synchformer was not finetuned due to computational constraints, we expect that a finetuned Stream-Synchformer would likely reflect the performance improvements seen with StreamSync and could potentially achieve near-perfect Acc^{tol1} metrics. However, our streaming methods significantly boost performance with only a minimal computational increase—StreamSync-LSTM adds just 3.8% to inference costs—while Synchformer requires an 824% increase in cost. Therefore, while large models like Synchformer advance research and are suitable for offline applications, their practicality for real-time uses such as broadcasting is limited. We recommend future research to focus on both high-performance and cost-effective AV Sync models, with practical solutions like APA and StreamSync providing a promising foundation.

	Acc	Acc^{tol1}	$R_{OC}A_{UC}$	mAP
Base	**0.566**	**0.841**	**0.959**	0.621
$K = 2$	0.547	0.831	0.959	0.613
$K = 4$	0.548	0.832	0.955	0.597
$H = 6$	0.564	0.834	0.958	**0.623**
$H = 18$	0.550	0.833	0.958	0.611

Table 5. Impact of training parameters K and H on performance.

5.2. Additional StreamSync Experiments

Additional experiments examined StreamSync's capabilities, particularly assessing architectural variations, training hyperparameters, and different inference settings. We began by evaluating the impact of substituting our SparseSync model's Transformer-based Streaming Head with an LSTM-based version of equivalent dimensionality. This transition is essential for broadcast applications where computational efficiency is critical. LSTMs not only provide computational advantages but also integrate seamlessly into the Streaming framework, requiring just the latest hidden state be stored and updated with a single Streaming Token at each AV synchronization check. As shown in Tab. 3, this modification actually enhances performance, with LSTMs showing a 1.1% increase in mAPs.

We also explored the influence of Window Hop Size on StreamSync's efficacy, as illustrated in Fig. 4. Training with 1-second hops and extending to 4 seconds during inference improved the performance of both StreamSync and APA SparseSync. Notably, StreamSync benefits more from larger hop sizes, demonstrating its superior capability to leverage broader contextual information. This advantage extends to streaming histories of up to 50 windows with 2-second hops, where StreamSync consistently outperforms APA, as evidenced in Tab. 4. These findings underscore the strengths of temporal streaming methods, particularly those like StreamSync that utilize learned components.

Lastly, we evaluated various training hyperparameters for StreamSync, focusing on the maximum number of Streaming Tokens, H, and the number allowed for full model backpropagation, K. The results, detailed in Tab. 5, suggest that $K = 3$ is optimal, with performance declining at $K = 2$. Our analysis indicates that extensive streaming histories during training are unnecessary; $H = 6$ slightly outperforms the standard parameters (K=12). Further increases in H or K did not yield improvements, suggesting diminishing returns with longer training histories.

5.3. Performance Analysis

We conduct a dataset analysis to understand the factors influencing StreamSync performance, initially focusing on the role of talking heads in videos. Building on prior research that highlights synchronized speech and lip movements as key cues for audio and video stream alignment [3, 6–8, 16], we categorize our test set into three scenarios

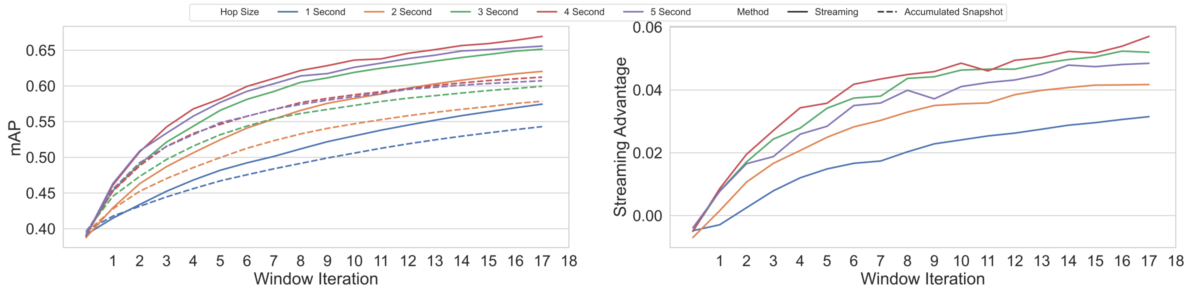

Figure 4. Impact of varying Window Hop Sizes for APA and StreamSync, with StreamSync increasingly improving over APA.

	Acc	Acc^{tol1}	$ROCAUC$	mAP
Talking-heads	0.706	0.980	0.986	0.750
Voiceovers	0.538	0.859	0.957	0.568
Others	0.358	0.610	0.876	0.411
Overall	0.567	0.845	0.961	0.632

Table 6. Evaluating the impact of talking heads for StreamSync (2-second hops and 18 streaming iterations). Comparative results for Pretrained SparseSync are included in the supplement.

based on talking head annotations. **Talking heads**: Videos with one or more visible speakers. **Voiceovers**: Videos where the visible individual is not the actual speaker, posing alignment challenges. **Others**: Videos not fitting the above categories, ranging from people-free scenes to those with non-speaking individuals. These categories do not account for the presence, or lack thereof, of other audio events. Tab. 6 shows that videos with visible speakers approach optimal performance, confirming visible speech synchronization as a strong signal for AV sync. Performance decreases in voiceover scenarios and is lowest in the ”Others” category, suggesting significant improvement opportunities.

We further analyze successful and unsuccessful synchronization instances across these scenarios in Section B of the supplementary material. For talking heads, typical failure modes include scenes with multiple potential speakers where the model cannot identify the active speaker, and instances where rhythmic head movements confound synchronization efforts. In contrast, in the ”Others” category, the model effectively uses sparse visual cues, like a golf ball strike or a scene change, for accurate predictions. This shows StreamSync’s sensitivity to both dense and sparse synchronization signals, enhancing its versatility. However, it struggles with continuous, subtle visual movements, such as those in musical performances with finger movements.

Voiceover scenarios highlight critical model weaknesses, notably false negatives when the talking head is adversely positioned or obscured. Furthermore, true voiceover situations often result in misinterpretations of visible synchronization cues, leading to inaccuracies. These findings underscore the need for advanced techniques to enhance accuracy in complex AV scenarios, and their inclusion in RealSync emphasizes its values for such improvements.

6. Conclusion

We have introduced RealSync, a comprehensive and diverse dataset tailored for training and evaluating AV Sync models under a variety of real-world conditions. This dataset, characterized by its wide range of content and synchronization scenarios, provides a solid foundation for developing and testing models adept at handling audio-visual misalignments across multiple contexts. Leveraging this resource, we developed StreamSync, a novel model that not only advances the state-of-the-art in AV Sync but also introduces temporal streaming capabilities, significantly enhancing adaptability to common synchronization challenges encountered in real-world applications.

While our contributions mark significant strides in addressing synchronization discrepancies, they are not without limitations. A notable challenge is the model’s performance in environments with unstable latency or changing synchronization states, both of which are common in real-world applications. The variability of network conditions can affect the synchronization state resulting in variable offsets. Future studies should focus on enhancing StreamSync’s adaptability to fluctuating latencies and exploring its response to abrupt synchronization changes, which could further broaden its practicality if the evaluations are well designed to simulate the complexities of real conditions.

Looking ahead, it is imperative to continue refining AV Sync technologies. Future work should explore extending the principles behind StreamSync to develop new models that incorporate streaming capabilities and are capable of utilizing even broader and more diverse datasets. Moreover, ongoing efforts to expand and diversify RealSync will be crucial. Such enhancements will ensure that it remains a valuable resource, providing an even more robust platform for developing sophisticated models that can precisely and adaptively manage the complexities of av synchronization.

References

[1] Rec. itu-r bt.1359-1 1 recommendation itu-r bt.1359-1 relative timing of sound and vision for broadcasting. 1998. 3

[2] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *Proceedings of the European conference on computer vision (ECCV)*, pages 435–451, 2018. 2

[3] Honglie Chen, Weidi Xie, Triantafyllos Afouras, Arsha Nagrani, Andrea Vedaldi, and Andrew Zisserman. Audio-visual synchronisation in the wild. *arXiv preprint arXiv:2112.04432*, 2021. 1, 2, 3, 7

[4] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 721–725. IEEE, 2020. 1, 2, 3, 4

[5] Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, and Furu Wei. Beats: Audio pre-training with acoustic tokenizers. *arXiv preprint arXiv:2212.09058*, 2022. 3

[6] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman. Lip reading sentences in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 3, 7

[7] J. S. Chung and A. Zisserman. Lip reading in profile. In *British Machine Vision Conference*, 2017. 1, 2, 3, 7

[8] Joon Son Chung and Andrew Zisserman. Out of time: automated lip sync in the wild. In *Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II 13*, pages 251–263. Springer, 2017. 1, 2, 3, 7

[9] Soo-Whan Chung, Joon Son Chung, and Hong-Goo Kang. Perfect match: Improved cross-modal embeddings for audio-visual synchronisation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3965–3969. IEEE, 2019. 1, 2

[10] Joshua P Ebeneze, Yongjun Wu, Hai Wei, Sriram Sethuraman, and Zongyi Liu. Detection of audio-video synchronization errors via event detection. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4345–4349. IEEE, 2021. 2, 3

[11] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780, 2017. 2, 3

[12] Akash Gupta, Rohun Tripathi, and Wondong Jang. Modeformer: Modality-preserving embedding for audio-video synchronization using transformers, 2023. 2

[13] Tavi Halperin, Ariel Ephrat, and Shmuel Peleg. Dynamic temporal alignment of speech to lips, 2018. 2

[14] J A Hanley and B J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, Apr. 1982. 6

[15] John Hershey and Javier Movellan. Audio vision: Using audio-visual synchrony to locate sounds. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. 2

[16] Vladimir Iashin, Weidi Xie, Esa Rahtu, and Andrew Zisserman. Sparse in space and time: Audio-visual synchronisation with trainable selectors. *arXiv preprint arXiv:2210.07055*, 2022. 1, 2, 4, 5, 7

[17] Vladimir Iashin, Weidi Xie, Esa Rahtu, and Andrew Zisserman. Synchformer: Efficient synchronization from sparse cues, 2024. 2, 4

[18] Venkatesh S. Kadandale, Juan F. Montesinos, and Gloria Haro. Vocalist: An audio-visual synchronisation model for lips and voices, 2022. 2

[19] Naji Khosravan, Shervin Ardeshir, and Rohit Puri. On attention modules for audio-visual synchronization. In *CVPR Workshops*, pages 25–28, 2019. 2

[20] You Jin Kim, Hee Soo Heo, Soo-Whan Chung, and Bong-Jin Lee. End-to-end lip synchronisation based on pattern classification. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 598–605, 2021. 2

[21] Kazuaki Kishida. *Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments*. National Institute of Informatics Tokyo, Japan, 2005. 6

[22] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. *Advances in Neural Information Processing Systems*, 31, 2018. 2

[23] Zongyi Joe Liu, Devin Chen, Yarong Feng, Yuan Ling, Shunyan Luo, Shujing Dong, and Bruce Ferry. Detect audio-video temporal synchronization errors in advertisements (ads). In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 16–22, 2022. 2

[24] KR Prajwal, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. 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*, pages 484–492, 2020. 1

[25] K R Prajwal, Rudrabha Mukhopadhyay, Vinay P. Namboodiri, and C.V. Jawahar. 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*, MM '20. ACM, Oct. 2020. 2

[26] Malcolm Slaney and Michele Covell. Facesync: A linear operator for measuring synchronization of video facial images and audio tracks. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000. 2

[27] Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Lip reading sentences in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6447–6456, 2017. 1, 2

[28] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 3

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 2