SILENCE: Protecting privacy in offloaded speech understanding on wimpy devices

Abstract

Speech serves as a ubiquitous input interface for embedded mobile devices. Cloud-based solutions, while offering powerful speech understanding services, raise significant concerns regarding user privacy. To address this, disentanglement-based encoders have been proposed to remove sensitive information from speech signals without compromising the speech understanding functionality. However, these encoders demand high memory usage and computation complexity, making them impractical for resource-constrained wimpy devices. Our solution is based on a key observation that speech understanding hinges on long-term dependency knowledge of the entire utterance, in contrast to privacy-sensitive elements that are short-term dependent. Exploiting this observation, we propose `SILENCE`, a lightweight system that selectively obscuring short-term details, without damaging the long-term dependent speech understanding performance. The crucial part of `SILENCE` is a differential mask generator derived from interpretable learning to automatically configure the masking process. We have implemented `SILENCE` on the STM32H7 microcontroller and evaluate its efficacy under different attacking scenarios. Our results demonstrate that `SILENCE` offers speech understanding performance and privacy protection capacity comparable to existing encoders, while achieving up to 53.3× speedup and 134.1× reduction in memory footprint.

1 Introduction

Privacy concern for cloud speech service The volume of speech data uploaded to the cloud for spoken language understanding (SLU) is steadily increasing [1, 12, 2], particularly in ubiquitous wimpy devices where textual input is inconvenient [41, 17, 3], e.g., home automation devices [32], smartwatches [37], telehealth sensors [22] and smart factory sensors [29] . However, exposing raw speech signal to the cloud raises privacy concerns [42]. It was revealed that contractors regularly listened to confidential details in Siri recordings to improve its accuracy [4]. This included private discussions, medical information, and even intimate moments.

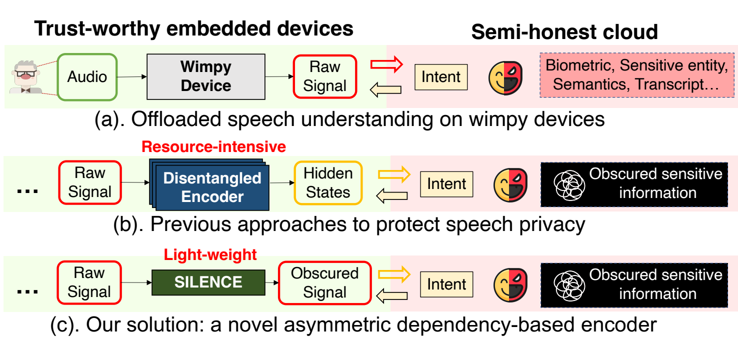


Figure 1: Illustration of offloaded speech understanding on wimpy devices and its privacy protection.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

There are many aspects of potential privacy leakage in cloud-based SLU. Among them: biometric or contextual privacy leakage have been well studied and somewhat solved by removing information relevant to such tasks without compromising the SLU accuracy [18, 35]; transcript protection (especially sensitive entities) is more challenging since it is deeply entangled with the SLU task itself. As shown in Figure 1, this paper focus on ensuring that cloud-based systems could efficiently classify the intent of SLU task (e.g., scheduling appointments or controlling home devices) while refraining from identifying the concrete entities (e.g., unintended names or passwords) in the spoken utterance, i.e., high word error rate (WER) of Automatic Speech Recognition (ASR) task. This is also a setting commonly used in speech privacy protection [44, 10, 16, 42, 15].

Prior approaches A prevalent method for private speech processing is employing *encoders*[1] based on disentanglement representation learning [44, 10, 28, 34], as illustrated in Figure 1(b). Those encoders extract the speech representations using pre-trained acoustic models, e.g., wav2vec [40, 10], conformer [26, 34] and Preformer [20, 44]. Furthermore, they promote representation disentanglement through adversarial training [25]. For example, PPSLU [44] uses a 12-layer transformer-based Preformer as its encoder.

As a result, disentanglement-based encoders still demand considerable computational resources, often exceeding tens of GFLOPs, to achieve effective disentanglement [11]. They are also memory-intensive, often comprising tens of millions of parameters. Consequently, they are unsuitable for embedded devices with limited memory. Moreover, it takes time-consuming adversarial training to disentangle the encoded representation for each specific SLU task. This aspect limits the flexibility and scalability for emerging SLU tasks. More motivating details will be presented in §2.2.

In this paper, we aim to achieve the real-time, privacy-preserving offloading of speech understanding task on wimpy devices like STM32H7 microcontroller [5] with only 1MB RAM. This goal necessitates a novel encoder design that must be both lightweight and effective in filtering out sensitive information, as illustrated in Figure 1(c).

Our solution We therefore present `SILENCE`, a **SI**mp**L**e **ENC**od**E**r designed for efficient privacy-preserving SLU offloading. It is based on the *asymmetric dependency* observation: SLU intent extraction (e.g., scenario identification) typically requires only long-term dependency knowledge across the entire utterance, while ASR task (e.g., recognizing individual words or phrases) needs short-term dependency, as confirmed by our experiments in §3.1. Based on it, `SILENCE` strategically partitions the utterance into several segments, selectively masking out the majority to enhance privacy by obscuring short-term details, without significantly damaging the long-term dependencies. The processed audio waveform is then transmitted to the cloud for SLU intent analysis. Additionally, we integrate a differential mask generator, inspired by interpretable learning methods [19], to optimize performance by automatically identifying how many and which segments to mask.

Results We deploy `SILENCE` on the STM32H7 microcontroller [5] and assess its performance using the SLURP dataset [13] in both black-box and white-box attack environments. `SILENCE` achieves 81.2% intent classification accuracy on SLURP, surpassing previous privacy-preserving SLU systems by up to 8.3%. Regarding privacy protection, `SILENCE` offers comparable security to earlier systems, with a word error rate of up to 81.6% and an entity error rate of 90.7% under malicious ASR attacks. Even against white-box attacks, where attackers are strongly assumed to have the same encoder structure and weights as `SILENCE`, plus partial data from malicious clients, `SILENCE` maintains 67.3% word error rate and 64.3% entity error rate. Additionally, `SILENCE` proves to be resource-efficient and feasible for wimpy devices, using only 394.9KB of memory and taking just 912.0ms to encode a 4-second speech signal. Integrated with RPI-4B for a fair comparison, `SILENCE` uses up to 134.1× less memory and operates up to 53.3× faster than prior systems. The accuracy of `SILENCE` is only 7% lower than unprotected SLU systems.

Contribution We have made the following contributions.

- Based on the observation of asymmetric dependency between SLU and ASR tasks, we propose `SILENCE`, a simple yet effective encoder system for privacy-preserving SLU offloading.

[1]Note that these encoders are not specifically transformer encoders; rather, they can be implemented using any NNs to encode speech signals.

- We are the first to retrofit interpretable learning methods to automatically configure the masking process for a better balance between privacy and utility in speech understanding tasks.
- We evaluate SILENCE on a wimpy microcontroller unit and demonstrate its effectiveness under various attack scenarios.

2 Related Work and Background

2.1 Privacy-preserving SLU

Spoken Language Understanding (SLU) is a critical component of modern voice-activated systems, responsible for interpreting human speech and translating it into structured, actionable commands. For instance, when a user says, "Set a meeting for tomorrow at 10 AM," the SLU system might map this to a structured intent such as `{scenario: Calendar, action: Create_entry}`.

Evolution of SLU Systems The evolution of SLU systems has seen a shift from traditional two-component systems, comprising ASR and Natural Language Understanding (NLU), to modern end-to-end neural networks [39, 27]. These advanced systems bypass the intermediate textual representation and directly map speech signals to their semantic meaning, enhancing efficiency and reducing error propagation. A typical end-to-end SLU model features an encoder, often with convolution and attention-based elements, and a decoder, including a transformer decoder and a connectionist temporal classification decoder. Many SLU systems incorporate encoders from pre-trained ASR models like HuBERT [45], replacing the original ASR decoder with one tailored for SLU tasks.

Threat Model Our threat model aligns with prior work [44, 10] where users (the victims) actively offloads their audio data to the cloud server (the adversary) for intended SLU tasks. Upon receiving the data, the adversary may employ automatic speech recognition to transcribe the audio and identify private entities [16, 42, 15]. Note that the transcriptions are often exceedingly detailed, containing much more information than the users intend to disclose. The goal of this paper is to ensure that the victims can reliably obtain the predefined SLU intent from the adversary, while preserving the adversary from discerning sensitive details or private entities in the transcript.

For instance, home pods might capture recordings of confidential daily interactions alongside explicit commands, presenting a paradigmatic case for SILENCE. Without SILENCE, over 80% of our private daily conversations could be automatically recognized and stored for unforeseen usage as will be analyzed in §5.1.

2.2 Inefficiency of Existing Approaches

Privacy-preserving methods Crypto-based approaches, such as HE [48] and MPC [24], have been proposed to provide encrypted computation. Unfortunately, they are technically slow and thus impractical for deployment on wimpy audio devices due to the significant increase in computation and communication complexity. For example, MPC-based PUMA [21] takes 5 minutes to complete one token inference, which is far too slow for real-time. Voice conversion is another method to protect speech content. Pr$\varepsilon\varepsilon$ch [9] integrates voice conversion with GPT-based generated noise protect privacy, but it is far from feasible for deployment on wimpy devices. Traditional peripheral devices, such as ultrasonic microphone jammers (UMJ), are designed to obscure raw speech by inserting non-linearity noise, thereby preventing illegal eavesdropping[23, 15]; however, they also corrupt speech semantics as well. A emerging and prevailing strategy is disentangling-based encoders [10, 44, 28]; they aim to create a disentangled and hierarchical representation of the speech signal devoid of sensitive data. But we reveal their performance issue next.

We conduct preliminary experiments to measure the resource consumption of the disentangling-based encoder of a pre-trained SLU model on a Raspberry Pi 4B (RPI-4B) [6] and Jetson TX2 (TX2) [7]. Our key observation is that disentangling-based privacy-preserving SLU system is too resource-intensive for practical deployment. As illustrated in Figure 2, a disentanglement encoder consumes 648.7MB memory and 12.8s for complete one inference on RPI-4B. Even in the strong TX2 with GPU, the encoder still takes 593.0ms to complete one inference. Considering the network latency, the end-to-end latency of the disentangling-based SLU offloading system only saves 0.7%

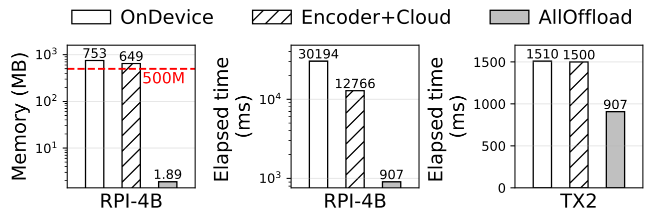


Figure 2: Cost of disentangling-based encoders [44] for a 4-second audio inference.

wall-clock time compared to the `OnDevice` inference without offloading, with a similar memory footprint over 500M.

Implications Disentangling-based encoders is slow and memory-intensive due to the complex encoder structure designed to separate sensitive information from the speech signal. Given the limited resource of wimpy devices, it is not practical for common privacy-preserving SLU scenarios. To enable practical privacy-preserving SLU, the encoder structure and the inference process need to be simplified.

3 `SILENCE` Design

3.1 System Design and Rationales

We introduce `SILENCE` to efficiently scrub raw audio for privacy-preserving SLU, as depicted in Figure 3. The key idea of `SILENCE` is simple and novel: it masks out a portion of audio segments before sending them to the cloud for SLU tasks. This design is based on an unique observation shown in Figure 4(c): when a portion of audio segments is masked out, the ASR model becomes incapable to recognize the phonemes in the masked frames, while the SLU model can still recognize the intent.

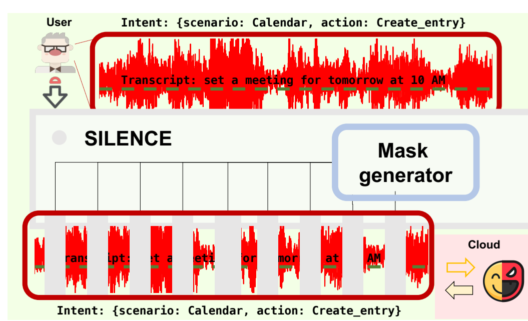
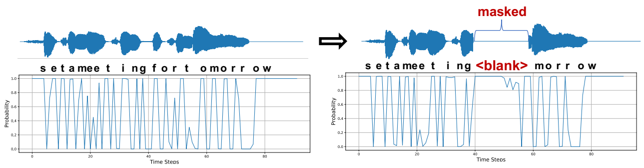


Figure 3: `SILENCE` overview. Red hard line represents the long-term dependency, while the green dotted line represents the short-term dependency.

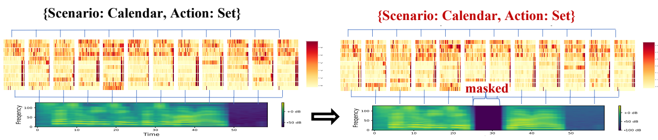
Design rationale Why is `SILENCE` able to protect the sensitive entity privacy while maintaining SLU accuracy? This capability is rooted in the *asymmetrical dependency* between the ASR and SLU task.

Speech is composed of many meta phonemes, and the generation of a single meta phoneme depends on its adjacent frame [42]. *Dependency* is defined as the length of frame that a model's output depends on. Figure 4(a) shows each phoneme is mainly dependent on a few frames, indicating short-term dependency. This phenomenon is referred to as "peaky behavior" in the ASR literature [47]. In contrast, an SLU model utilizes an attention-based decoder [45] to capture the relationship between the entire utterance and the intent, implying that the intent is long-term dependent on the whole utterance.

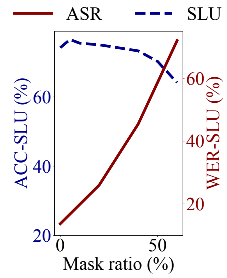
Formally, `SILENCE` is a simple encoder based on asymmetrical dependency-based masking. This simple masking encoder is defined as: $\hat{x} = x \odot \mathbb{Z}$, where x is the input audio signal, $\odot$ represents the element-wise multiplication, $\hat{x}$ is the masked audio signal and $\mathbb{Z}$ is the binary masking vector with the same dimension as x. $\mathbb{Z}$ consists of k uniform portion, with all 0s or 1s in one portion



(a) Peaky phoneme is short-dependent



(b) Attention seeks intent globally



(c) Empirical performance under different ratios of masked portion.

Figure 4: Foundation of `SILENCE`: asymmetrical dependency. (a). ASR task is short-term dependent on the peaky phoneme probability. (b). SLU task is long-term dependent on knowledge from the whole utterance. (c). Empirical results.

to mask-out or preserve the complete adjacent frames, respectively. This simple encoder forms the basis of `SILENCE`'s efficiency and privacy-preservation capacity, enabling secure offloading of speech understanding tasks on wimpy devices.

The configuration challenges: Figure 4(c) demonstrates that the ratio of masked portion plays a crucial role in balancing the privacy (WER-ASR) and utility (ACC-SLU). Currently, `SILENCE` employs a trivial masking mechanism, necessitating clients to undertake a time-intensive hyper-parameter adjustment about the extent and location of masking. Incorrect masking configurations can result in significant loss of global long-term dependency, negatively affecting SLU accuracy, or insufficient masking of sensitive information, thus compromising privacy. Therefore, we face critical questions: how many and which portions should be masked?

3.2 Online Configurator for `SILENCE`

To address these challenges, we derive a differential mask generator from the interpretable learning [19] as a online configurator for `SILENCE`. This automatically generate the masking vector $\mathbb{Z}$. The mask generator is trained to identify how many and which portions to mask, optimizing the privacy-utility balance.

Differentiable mask generator The configurator model aims to minimize the discrepancy between masked and original output by generating a mask $\mathbb{Z}$. Formally, we define the number of unmasked portions as $\mathcal{L}_0$ loss:

$$\mathcal{L}_0(\phi, x) = \sum_{i=1}^{n} \mathbf{1}_{[\mathbb{R}_{\neq 0}]}(\mathbb{Z}_i) \tag{1}$$

where ϕ is the mask generator, $\mathbf{1}(\cdot)$ is the indicator function. We minimize $\mathcal{L}_0$ for dataset $\mathcal{D}$, ensuring that predictions from masked inputs resemble those from the origin model:

$$\min_{\phi} \sum_{x \in \mathcal{D}} \mathcal{L}_0(\phi, x) \tag{2}$$

$$\text{s.t. } \mathrm{D}_\star[y \| \hat{y}] \leq \gamma \quad \forall x \in \mathcal{D} \tag{3}$$

where $\hat{y} = f(\hat{x}), y$ is the tokenized label, $\mathrm{D}_\star[y\|\hat{y}]$ is the KL divergence and the margin $\gamma \in \mathbb{R}_{>0}$ is a hyperparameter.

Given that $\mathcal{L}_0$ is discontinuous and has zero derivative almost everywhere, and the mask generator ϕ requires a discontinuous output activation (like a step function) for binary masks, we utilize a sparse relaxation to binary variables [30, 14] instead of the binary mask during training.

Holistic workflow As shown in Figure 5, `SILENCE` encompasses two phases:

(1) *Offline phase*: (**1a**) First, `SILENCE` trains a differentiable mask generator. The client selects a mask generator model, potentially a submodule of a pre-trained ASR model, such as HuBERT's

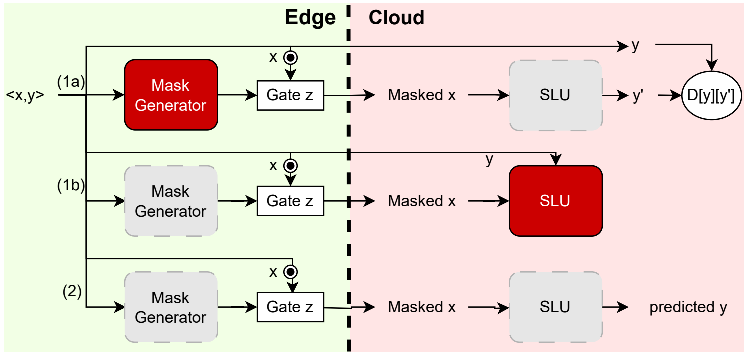


Figure 5: `SILENCE` workflow. (1) *Offline phase*: (**1a**) Training mask generator and (**1b**) adapting cloud SLU model to it; (2) *Online phase*: Conducting could inference with the masked x. Only masked input audio x and insensitive intent label y are exposed to the cloud.

CNN feature extractor. A small gate model is then integrated with this submodule. The combined model processes the input audio and generates a mask. This mask selectively conceals parts of the input, ensuring retention of only vital SLU information while hiding sensitive data. The masked input is then forwarded to either a trusted cloud service or a local SLU model for obtaining masked output. The mask generator is fine-tuned to minimize the discrepancy between the masked output logits and the original intent, as defined in Equation (1-3).

(**1b**) Second, `SILENCE` adapts the cloud model . Here, the client forwards the masked input and a specific SLU intent (e.g., "set alarm") to the cloud-based SLU model. The model undergoes fine-tuning to adapt to the masked inputs. This process includes adjusting the model parameters for accurate recognition and response to SLU commands based on the masked input.

(2) *Online phase*: In online speech understanding, the client sends the masked input to the cloud SLU model. Using the adapted model, the cloud-based SLU accurately identifies and executes the intended SLU action or response.

Configurator cost analysis Training the differentiable mask generator is affordable for the client. Our experiments indicate that convergence is achieved with approximately 200 audio samples, equivalent to 600 seconds of audio. This process takes up to 30 seconds on an A40 GPU. Adapting the SLU model to each mask generator is a one-pass effort. This adaptation is relatively trivial, especially when starting from a fine-tuned SLU model rather than building from scratch. This aspect of the process incurs minimal cost compared to the training of the cloud SLU model. Moreover, these costs can be amortized over a large number of edge users in the long run, making it an economically viable solution.

Remark Note that the mask generator is not developed for tagging sequences at a semantic level. Rather, its design focuses on identifying segments that are more relevant to the SLU task. This task is essentially a relatively straightforward binary classification problem, which is proven to be effective in prior interpretable learning literature [19, 14] and light-weight enough for real-time inference.

4 Implementation and Methodology

We have fully implemented the `SILENCE` prototype atop SpeechBrain [38], a PyTorch-based and unified speech toolkit. As prior work [45], we use SpeechBrain to train the differential mask generator and simulate the cloud training process. After that, we deploy the trained mask generator into the embedded devices and evaluate the end-to-end performance.

Hardware and environment Offline training is simulated on a server with 8 NVIDIA A40 GPUs. The trained mask generator is deployed into the STM32H7 [5] or Raspberry PI 4 (RPI-4B) [6]. STM32H7 is a wimpy microcontroller with 1MB RAM. RPI-4B is a popular development board with 4GB RAM. We embed the approaches not feasible to fit in the STM32H7 into the RPI-4B.

Models We design four types of mask generator structures: (1) `Random`: a random binary vector generator with 50% portion masked; (2) `SILENCE-S`: a learnable mask generator with only one MLP gate; (3) `SILENCE-M`: a learnable mask generator with one HuBERT encoder layer and the gate; (4) `SILENCE-L`: a learnable mask generator with three HuBERT encoder layers and the gate. As for the cloud SLU model, we simulate it using the SoTA end-to-end SLU model [45]. It replaces the ASR decoder of pre-trained HuBERT with SLU attentional decoder.

Dataset and Metrics We run our experiments on SLURP [13] with 102 hours of speech. SLURP's utterances are complex and closer to daily human speech. We select scenario classification accuracy to measure the SLU understanding performance (ACC-SLU). Following prior work [44], we choose large-scale English reading corpus LibriSpeech [33] for a multi-task protection scenario. In the multi-task protection scenario, not only the SLU command utterance (SLURP) but also the background or the subsequent utterance (LibriSpeech) are uploaded to the cloud. WER is used to measure the attack performance. More specifically, we utilize WER-SLU to measure the attacker's capacity to recognize the word information in the uploaded SLU audio itself, and WER-ASR as the WER of recognized accompanying audio, i.e., LibriSpeech dataset. We also report the private entity recognition error rate (EER) to ensure that the cloud model is not able to recognize the private information in the speech signal. As for latency, we sequentially fed test audios into the local model without any window processing[2] and recorded the average forward time as the local execution time.

Baselines We compare `SILENCE` to the following alternatives: (1) `OnDevice` means the cloud SLU model is downloaded and run locally on the client device. (2) `AllOffload` means the raw audio is uploaded to the cloud for SLU inference. (3) `VAE` [10] is the vanilla variational auto-encoder method that uses adversarial training to disentangle the private information from speech signal. (4) `PPSLU` [44] is the state-of-the-art disentangling-based SLU privacy-preserving system, which uses 12 transformer layers to separate the SLU information into a part of the hidden layer and only sends those hidden layers to the cloud for SLU inference.

Attack scenarios. We use three attacks encompassing both black-box and white-box attacks: (1) `Azure` represents a black-box attacker scenario, in which the masked audio is transmitted to Azure [31] for automatic speech recognition. (2) `Whisper` simulates a SoTA cloud-based ASR model. This black-box attacker uses the pre-trained $Whisper.medium.en$ model [36], directly downloaded from HuggingFace [46]. (3) `Whisper(White-box)` constitutes a white-box attack. Here, we hypothesize that certain users are malicious and disclose the mask generator's structure and weights, along with their own audio data, to the `Whisper` attack model. `Whisper(White-box)` then utilizes this collected data from malicious users to adapt the pre-trained $Whisper.medium.en$ model to the specific masking pattern.

Hyper-parameters During the offline phase in Figure 5, we use the Adam optimizer with a learning rate of 1e-5 and a batch size of 4. For the inference step, we use the batch size of 1 to simulate the real streaming audio input scenario. The end-to-end cloud SLU latency is measured by invoking Azure APIs following previous work [43]. KL threshold λ is set as 0.15 for all mask generators. Attack model is set as `Whisper` without special declaration.

5 Evaluation

5.1 End-to-end performance

`SILENCE` **achieves comparable accuracy performance and privacy protection capacity to previous encoders.** As shown in Figure 6, we compare the accuracy of `SILENCE` with all baselines. `OnDevice` offloads no signals to the cloud and thus has the best privacy protection (WER=100). It is observed that `SILENCE` could achieve up to 81.1% accuracy, with less than 7% accuracy loss compared to unprotected `AllOffload` and local `OnDevice` SLU model. Its rationale is that we mainly mask the short-dependent frames that does not significantly affect the SLU performance. We also compare the performance of `SILENCE` with the SoTA privacy-preserving SLU system, i.e., `PPSLU` [44]. `SILENCE` achieves 7.2% higher accuracy than `PPSLU` which tries to apply complex non-linear transformation to the hidden layer to prevent malicious re-construction, but this might also damage part of the SLU information. In terms of privacy preservation, our learnable mask generator achieves up to 78.6% WER using `SILENCE-L`, indicating a privacy-preserving capacity on par with

[2]The average duration of test SLU snippets is 2.8 seconds, with a maximum of 21.5 seconds, which is shorter than the maximum input window of speech models (e.g., 30 seconds for Whisper [36]).

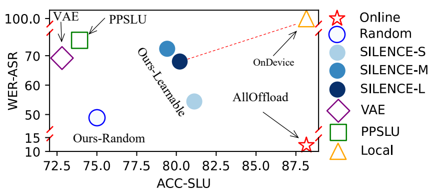


Figure 6: Performance of different privacy-preserving SLU approaches.

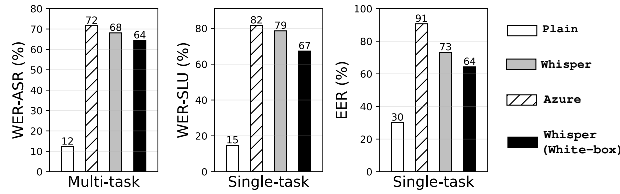


Figure 7: SILENCE privacy-preserving capacity under different attack models.

PPSLU. Furthermore, we complete the inference with much lower delays and memory footprint as will be shown in Figure 9.

SILENCE **is resistant to different attack models.** As illustrated in Figure 7, SILENCE increases the SLU-WER from 14.7% to 78.6% under the attack model Whisper. As for the online attack model Azure, SILENCE increases the SLU-WER from 14.7% to 81.6%. According to our returned service details, we find that over 50% of the sent audios are tagged as "$ResultReason.NoMatch$", which means audios are recognized as null utterances by the Azure ASR model. Whisper(White-box) is a white-box attack model, which means the attacker has the same mask generator structure and weights as the SILENCE. We still achieve more than 50% SLU-WER under this attack model. This is because even Whisper(White-box) is fine-tuned to fill some of the missing frames, it still could not recover the private missing frames. Because masking the short-dependent frames fundamentally destroys the raw audio signal. It is not possible to re-construct the phoneme without knowing any speech information. In the last subfigure, we show the high entity error rate to demonstrate that the private entity is not leaked.

SILENCE **scales to better privacy-accuracy trade-off with a larger mask generator.** We explore the impact of the threshold γ of SILENCE under different mask generator structures. As shown in Figure 8, the threshold γ controls the trade-off between the privacy and utility. When γ is small, the mask generator is more conservative, leading to higher the utility a lower the masking portion. As we have discussed in Section 3, a lower rate of masking portions leads to higher possibility of privacy entity leakage. When γ is large, the mask generator is more aggressive, enhancing privacy. Another way to achieve more practical privacy-utility balance is using a more complex mask generator structure, e.g., SILENCE-L. It achieves higher utility with the same privacy level compared to SILENCE-S, albeit with less efficiency, as shown in § 5.2.

5.2 System cost

SILENCE protects the private entities efficiently as shown in Figure 9. Different from prior encoders using complex disentanglement model, SILENCE only requires a light-weight mask generator to scrub the private information. The size of this generator varies according to different mask generator structures. For the smallest mask generator, SILENCE-S, it only requires a 394.9KB memory footprint, and could successfully embed into the wimpy STM32H7 with 2MB RAM. SILENCE is efficient not only in terms of memory footprint but also in latency. SILENCE-S completes the local encoding with only 912.2ms on the wimpy STM32H7. For a fair comparison, we embed SILENCE-S

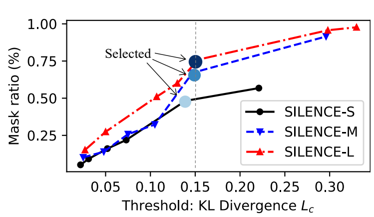
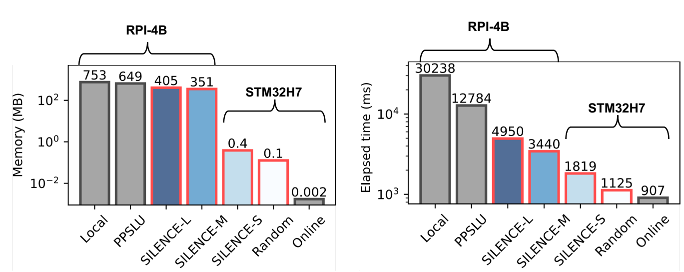


Figure 8: Effect of threshold with different mask generators



(a) Memory footprint (b) End-to-end latency

Figure 9: Comparison of resource cost in different SLU approaches. Ours are highlighted in red.

into RPI-4B and find that it is $18.1\times$ faster and $134.1\times$ less memory footprint than PPSLU. Even with the strong mask generator SILENCE-L, SILENCE achieves up to $7.5\times$ lower encoding latency and consumes $1.9\times$ less memory compared to OnDevice.

6 Conclusion and Discussions

SILENCE is an efficient and privacy-preserving end-to-end SLU system based on the asymmetrical dependency between ASR and SLU. SILENCE selectively mask the short-dependent sensitive words while retaining the long-dependent SLU intents. Together with the differentiable mask generator, SILENCE shows superior end-to-end inference speedup and privacy protection under different attack scenarios.

Limitations: While for the first time, SILENCE provides a feasible privacy-preserving solution for wimpy audio devices, it introduces a huge design space for mask generator structures. The mask generator is akin to a lock; a genius lock design can protect privacy in the smallest of spaces, but a poor lock design can be bulky and easily broken. In this work, we simply inherit the SLU model structure and instantiate three sub-models from it to demonstrate better efficiency than previous encoders. Researchers can explore other structures for a better privacy-accuracy-efficiency trade-off. We will open-source all the code and checkpoints to facilitate further research in this direction.

Some other potential limitations about lossy privacy-preserving capacity, the need for fine-tuning the cloud SLU model and the scope of defended threat model are thoroughly discussed below for further clarification.

Is current privacy-preserving capacity enough? The quantitative WER 80% is considered secure enough, as previous encoders have strived to reach that level [44, 10]. And some SLU transcripts contain the intent word, so the successfully inferred word might be a non-private intent word. For instance, in one test audio transcript, "I want some jazz music to play", the intent is 'scenario': 'play', 'action': 'music'. The interpretation of the malicious cloud ASR, "all subjects were used to play", is acceptable since the predicted phrase "to play" contains no private information. This scenario is typical for most audios; we managed to preserve 90% of the private entities in Figure 6. This achievement matches the SoTA in privacy-preserving capacity, with up to $30\times$ lower latency and $100\times$ memory reduction.

Why and how to fine-tune the cloud SLU Model? Initially, the cloud SLU is a generic pre-trained speech model lacking the capability to accurately understand personalized user intent. It is crucial to fine-tune the cloud SLU for better personalized intent understanding[3]. Secondly, while short-dependent masking does not eliminate intent information, it does impact specific details within the attention map, as depicted in Figure 4(b). Fine-tuning the cloud SLU model helps mitigate this impact and enhances the understanding of the user's intent.

Currently, cloud service providers have already offered APIs that allow users to fine-tune their personalized cloud speech model. For example, Azure has introduced the Custom Speech service [8], which enables users to fine-tune the model for improved personalized outcomes. In this work, we simulate the tunable cloud model using the open-source model to perform more detailed analysis, such as different attacking scenarios

Could private semantic detection attack be prevented? SILENCE does not initially target private semantic detection attacks. For example, eavesdropping on specific financial words and political framing are *out-of-scope*. However, we can offer defense capabilities against them as discussed below. The mask generator, controlled by the user, is trained to scrub utterances unrelated to the public intent. Private entities not predefined by the user are almost never included in the masked audio. Therefore, even if an attacker possesses a well-defined semantic and the mask generator, training the detection threat model is challenging because the synthetic masked audio lacks clear representations of the private semantic. Consequently, though not initially designed for this purpose, our mask generators successfully discourage the malicious cloud provider from detecting private semantics.

[3]Note that a general speech model is sufficient for training the local mask generator in Figure 5 step (1a), as the focus is not on generating precise intent but rather on obtaining a coarse-grained distribution of numerical logits to facilitate mask generator training.

References

[1] https://openai.com/blog/chatgpt-can-now-see-hear-and-speak.

[2] https://huggingface.co/models?sort=downloads.

[3] https://safeatlast.co/blog/siri-statistics/.

[4] https://www.cnbc.com/2019/08/28/apple-apologizes-for-listening-to-siri-conversations.html.

[5] https://www.st.com/en/microcontrollers-microprocessors/stm32h7-series.html.

[6] https://www.raspberrypi.com/products/raspberry-pi-4-model-b/.

[7] https://developer.nvidia.com/embedded/jetson-tx2.

[8] https://azure.microsoft.com/en-us/blog/improve-speechtotext-accuracy-with-azure-custom-speech/.

[9] Shimaa Ahmed, Amrita Roy Chowdhury, Kassem Fawaz, and Parmesh Ramanathan. Pr$\varepsilon\varepsilon$ch: A system for privacy-preserving speech transcription. arXiv preprint arXiv:1909.04198 v2, 2019.

[10] Ranya Aloufi, Hamed Haddadi, and David Boyle. Privacy-preserving voice analysis via disentangled representations. In Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop, pages 1–14, 2020.

[11] Siddhant Arora, Siddharth Dalmia, Xuankai Chang, Brian Yan, Alan Black, and Shinji Watanabe. Two-pass low latency end-to-end spoken language understanding. arXiv preprint arXiv:2207.06670, 2022.

[12] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. Advances in neural information processing systems, 33:12449–12460, 2020.

[13] Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser. Slurp: A spoken language understanding resource package. arXiv preprint arXiv:2011.13205, 2020.

[14] Jasmijn Bastings, Wilker Aziz, and Ivan Titov. Interpretable neural predictions with differentiable binary variables. arXiv preprint arXiv:1905.08160, 2019.

[15] Yike Chen, Ming Gao, Yimin Li, Lingfeng Zhang, Li Lu, Feng Lin, Jinsong Han, and Kui Ren. Big brother is listening: An evaluation framework on ultrasonic microphone jammers. In IEEE INFOCOM 2022-IEEE Conference on Computer Communications, pages 1119–1128. IEEE, 2022.

[16] Peng Cheng and Utz Roedig. Personal voice assistant security and privacy—a survey. Proceedings of the IEEE, 110(4):476–507, 2022.

[17] Leigh Clark, Philip Doyle, Diego Garaialde, Emer Gilmartin, Stephan Schlögl, Jens Edlund, Matthew Aylett, João Cabral, Cosmin Munteanu, Justin Edwards, et al. The state of speech in hci: Trends, themes and challenges. Interacting with computers, 31(4):349–371, 2019.

[18] Trung Dang, Om Thakkar, Swaroop Ramaswamy, Rajiv Mathews, Peter Chin, and Françoise Beaufays. A method to reveal speaker identity in distributed asr training, and how to counter it. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4338–4342. IEEE, 2022.

[19] Nicola De Cao, Michael Schlichtkrull, Wilker Aziz, and Ivan Titov. How do decisions emerge across layers in neural models? interpretation with differentiable masking. arXiv preprint arXiv:2004.14992, 2020.

[20] Keqi Deng, Songjun Cao, Yike Zhang, and Long Ma. Improving hybrid ctc/attention end-to-end speech recognition with pretrained acoustic and language models. In 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pages 76–82. IEEE, 2021.

[21] Ye Dong, Wen-jie Lu, Yancheng Zheng, Haoqi Wu, Derun Zhao, Jin Tan, Zhicong Huang, Cheng Hong, Tao Wei, and Wenguang Cheng. Puma: Secure inference of llama-7b in five minutes. arXiv preprint arXiv:2307.12533, 2023.

[22] Lloyd E Emokpae, Roland N Emokpae, Wassila Lalouani, and Mohamed Younis. Smart multimodal telehealth-iot system for covid-19 patients. IEEE Pervasive Computing, 20(2):73–80, 2021.

[23] Ming Gao, Yike Chen, Yajie Liu, Jie Xiong, Jinsong Han, and Kui Ren. Cancelling Speech Signals for Speech Privacy Protection against Microphone Eavesdropping. Association for Computing Machinery, New York, NY, USA, 2023.

[24] Oded Goldreich. Secure multi-party computation. Manuscript. Preliminary version, 78(110):1–108, 1998.

[25] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. Advances in neural information processing systems, 27, 2014.

[26] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. arXiv preprint arXiv:2005.08100, 2020.

[27] Parisa Haghani, Arun Narayanan, Michiel Bacchiani, Galen Chuang, Neeraj Gaur, Pedro Moreno, Rohit Prabhavalkar, Zhongdi Qu, and Austin Waters. From audio to semantics: Approaches to end-to-end spoken language understanding. In 2018 IEEE Spoken Language Technology Workshop (SLT), pages 720–726. IEEE, 2018.

[28] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[29] Naveen Kumar and Seul Chan Lee. Human-machine interface in smart factory: A systematic literature review. Technological Forecasting and Social Change, 174:121284, 2022.

[30] Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through l_0 regularization. arXiv preprint arXiv:1712.01312, 2017.

[31] Microsoft. Azure asr. `https://azure.microsoft.com/en-us/products/ai-services/speech-to-text/`.

[32] Nombulelo CC Noruwana, Pius Adewale Owolawi, and Temitope Mapayi. Interactive iot-based speech-controlled home automation system. In 2020 2nd International Multidisciplinary Information Technology and Engineering Conference (IMITEC), pages 1–8. IEEE, 2020.

[33] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 5206–5210. IEEE, 2015.

[34] Cal Peyser, Ronny Huang Andrew Rosenberg Tara N Sainath, Michael Picheny, and Kyunghyun Cho. Towards disentangled speech representations. arXiv preprint arXiv:2208.13191, 2022.

[35] Jianwei Qian, Haohua Du, Jiahui Hou, Linlin Chen, Taeho Jung, and Xiang-Yang Li. Hide-behind: Enjoy voice input with voiceprint unclonability and anonymity. In Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems, pages 82–94, 2018.

[36] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In International Conference on Machine Learning, pages 28492–28518. PMLR, 2023.

[37] Joel M Raja, Carol Elsakr, Sherif Roman, Brandon Cave, Issa Pour-Ghaz, Amit Nanda, Miguel Maturana, and Rami N Khouzam. Apple watch, wearables, and heart rhythm: where do we stand? Annals of translational medicine, 7(17), 2019.

[38] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, et al. Speechbrain: A general-purpose speech toolkit. arXiv preprint arXiv:2106.04624, 2021.

[39] Subendhu Rongali, Beiye Liu, Liwei Cai, Konstantine Arkoudas, Chengwei Su, and Wael Hamza. Exploring transfer learning for end-to-end spoken language understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 13754–13761, 2021.

[40] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. arXiv preprint arXiv:1904.05862, 2019.

[41] Suranga Seneviratne, Yining Hu, Tham Nguyen, Guohao Lan, Sara Khalifa, Kanchana Thilakarathna, Mahbub Hassan, and Aruna Seneviratne. A survey of wearable devices and challenges. IEEE Communications Surveys & Tutorials, 19(4):2573–2620, 2017.

[42] Ke Sun, Chen Chen, and Xinyu Zhang. " alexa, stop spying on me!" speech privacy protection against voice assistants. In Proceedings of the 18th conference on embedded networked sensor systems, pages 298–311, 2020.

[43] Rongxiang Wang and Felix Lin. Efficient deep speech understanding at the edge. arXiv preprint arXiv:2311.17065, 2023.

[44] Yinggui Wang, Wei Huang, and Le Yang. Privacy-preserving end-to-end spoken language understanding. In Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, pages 5224–5232, 2023.

[45] Yingzhi Wang, Abdelmoumene Boumadane, and Abdelwahab Heba. A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding. arXiv preprint arXiv:2111.02735, 2021.

[46] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771, 2019.

[47] Albert Zeyer, Ralf Schlüter, and Hermann Ney. Why does ctc result in peaky behavior? arXiv preprint arXiv:2105.14849, 2021.

[48] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning. In 2020 USENIX annual technical conference (USENIX ATC 20), pages 493–506, 2020.

NeurIPS Paper Checklist

(1) **Claims**

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: Our contribution is outlined as a seperated paragraph in §1.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

(2) **Limitations**

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We thoroughly discuss the limitations of our work in §6.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

(3) **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

(4) **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide detailed instructions on how to reproduce the main experimental results in §4. We will open-source the code and data upon acceptance.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

(5) **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will open-source the code and data upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

(6) **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide detailed instructions on how to reproduce the main experimental results in §4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

(7) **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Error bars are not reported because of the time limit. We will attempt to add them in the camera-ready version.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

(8) **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide detailed hardware information in §4 and the intended runtime in §3.2 and §5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

(9) **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We have reviewed the NeurIPS Code of Ethics and believe that our research conforms to it.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

(10) **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We have discussed and provided real-world examples of both positive and negative societal impacts in §1 and §2.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

(11) **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper is intended for privacy protection and does not involve high-risk data or models.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

(12) **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have properly cited the original code, data and models in §4.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

(13) **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [No]

Justification: We will provide detailed documentation for the new assets upon acceptance.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

(14) **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

(15) **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

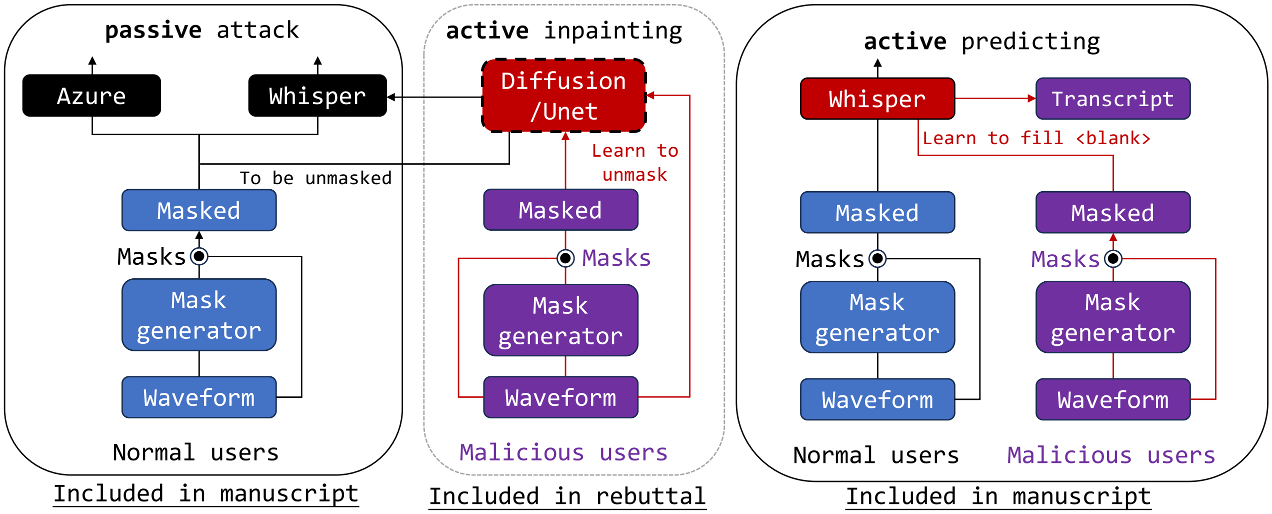


Figure 1: Mask generator and different attack scenarios, including both passive and active attacks.

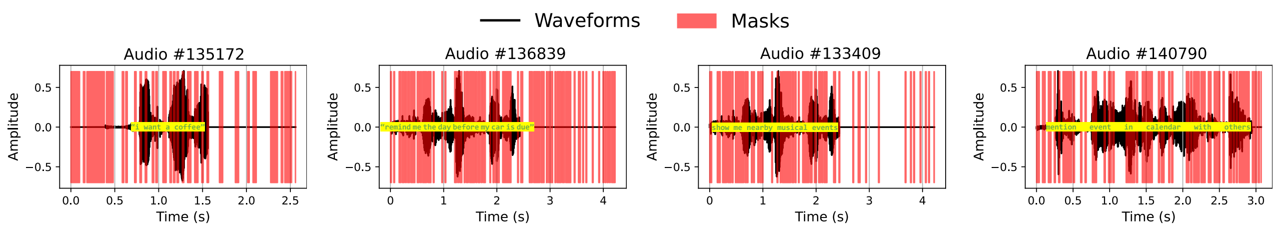


Figure 2: Illustration of the generated masks on audios selected randomly from SLURP. Local utterances are efficiently disrupted according to different transcripts patterns as highlighted within.

	PlainText	Azure	Naive Whisper	U-Net	CQT-Diff	Whisper predict (white box)
WER-SLU (%)	14.7	81.6	78.6	82.5	74.3	67.3
WER-ASR (%)	12.3	71.6	681.	71.4	65.9	64.4

Table 1: Potential attack Word Error Rate (WER) under different attack scenarios.

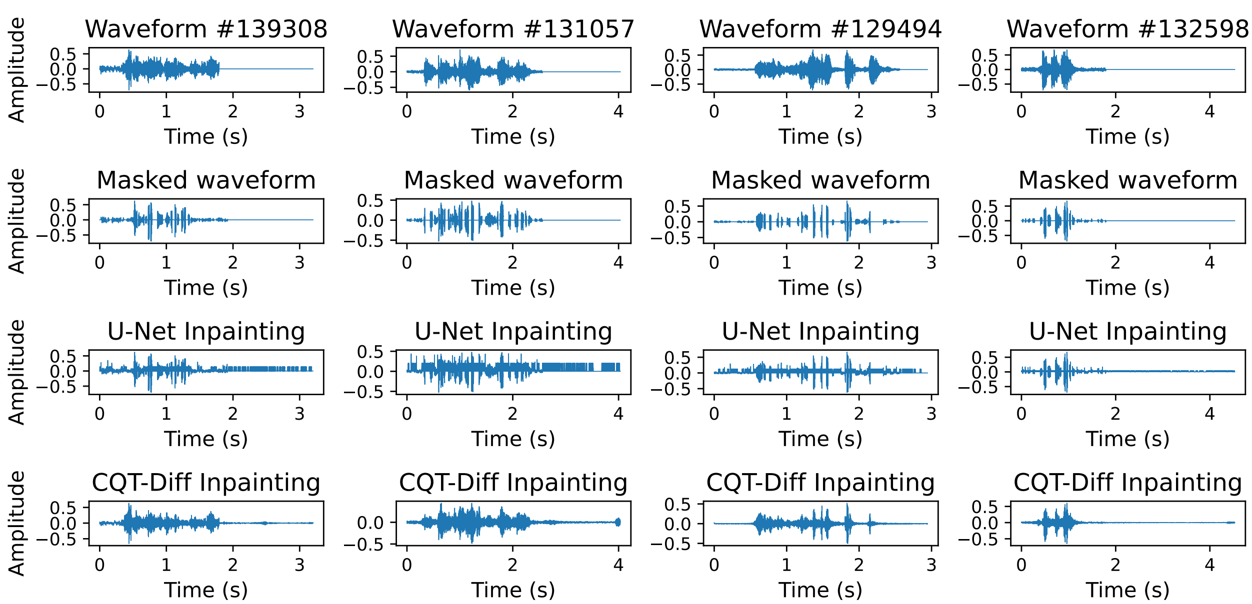


Figure 3: The reconstructed waveforms of different active inpainting attacks. Dataset: SLURP.