LFBA: Latent-Space Frame-Level Backdoor Attacks on Keyword Spotting Systems

Zexin Li[1], Wenhan Yao[1], Ye Xiao[1], Jinsu Yang[1], Zedong Xing[1], Xiarun Chen[2],
Fen Xiao[1], Weiping Wen[2,*]

Abstract— **Modern deep learning models increasingly rely on third-party data processing, exposing vulnerabilities to backdoor attacks. Existing audio backdoor methods often compromise stealthiness by introducing perceptible modifications. This paper proposes Latent-space Frame-level Backdoor Attacks (LFBA), a novel framework that manipulates frame-level features in latent space to achieve imperceptible and effective backdoor injection. Our approach extracts and transforms frame-level features to subtly alter rhythmic patterns, such as compressing or expanding temporal segments, without modifying semantic content or speaker characteristics. Evaluations demonstrate excellent attack effectiveness while maintaining near-original audio quality. Our attack evades human perception and automated detection, maintaining robustness even after defensive fine-tuning. This work reveals critical risks in outsourced speech model training and establishes a new paradigm for stealthy, latent-space poisoning in speech-controlled systems.**

I. INTRODUCTION

Speech-controlled systems, powered by Keyword Spotting (KWS) technology, have become ubiquitous in modern smart devices, enabling seamless human-computer interaction through voice commands [1]. However, the growing reliance on deep neural networks (DNNs) [2] and large-scale datasets [3], [4] for KWS tasks has precipitated a dilemma. The escalating computational demands of training sophisticated DNN models, coupled with the storage requirements of massive audio datasets, have driven many developers to outsource model training and data processing to third-party cloud platforms. This trend towards computational offloading, while expedient in reducing local computational load, creates critical attack surfaces in the deep learning pipeline. For instance, the centralized storage of sensitive speech data and proprietary models in external servers has made KWS systems particularly vulnerable to emerging security threats such as backdoor attacks. Backdoor attackers from third-party platforms with access to the training pipeline can surreptitiously manipulate the data or training procedure to embed hidden triggers, like subtle patterns or features, within the model through data poisoning. The resulting backdoored model behaves normally on clean samples—inputs without the trigger—maintaining its expected performance. However, when presented with poisoned samples—inputs deliberately crafted to include the embedded trigger—the model exhibits abnormal behavior, such as misclassifying inputs into a target class predetermined by the attacker. This duality of functionality makes backdoor attacks particularly insidious, as the compromised model passes conventional detection during testing or deployment, becoming a dormant weapon awaiting trigger activation.

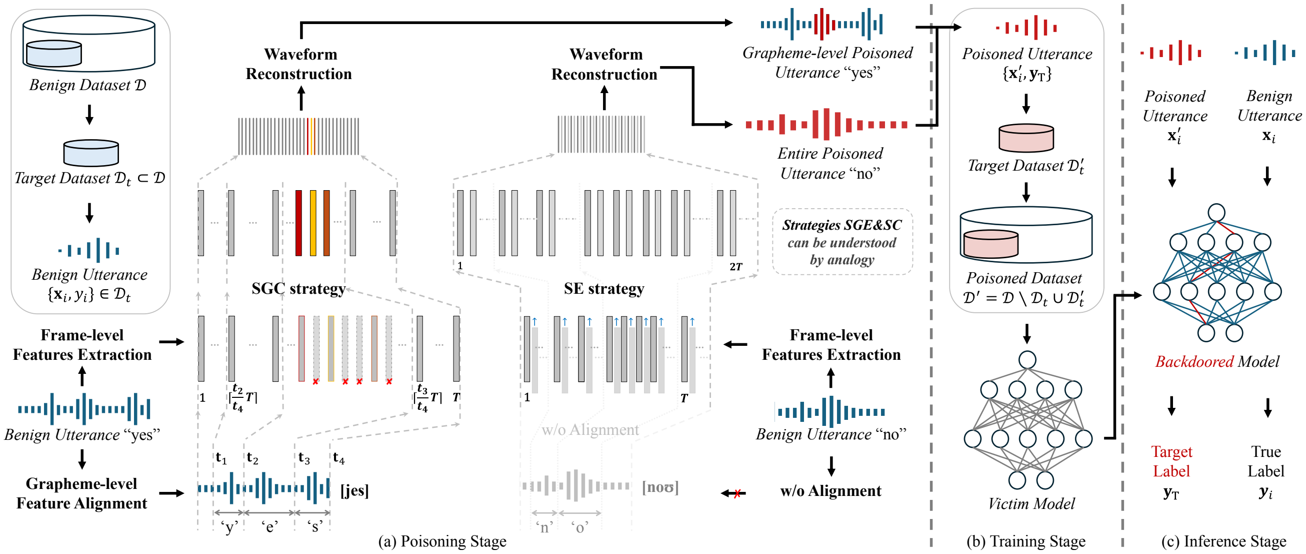
In the context of speech recognition, including KWS, existing backdoor attack methods primarily focus on manipulating the audio waveform or its spectrogram representation. These methods often involve adding triggers such as specific noise patterns [5], [6], [7], [8], modifying acoustic elements [9], [10], [11] or even altering phonetic components [12]. While these approaches have demonstrated the feasibility of backdoor attacks in speech systems, they often suffer from limitations in terms of stealthiness. The introduced triggers can sometimes be perceptible to human listeners or need complex dedicated deployment pipelines.

To overcome the limitations of prior work, we propose a novel backdoor attack framework operating directly within the latent feature space of speech representations, called Latent-space Frame-level Backdoor Attacks (LFBA). Our key insight is to leverage a powerful self-supervised learning (SSL) model that excels at extracting fine-grained, frame-level features. Building on this foundation, we design feature transformation strategies, namely frame compression and frame expansion, which subtly alter the temporal dynamics within the latent space, manifesting as slight rhythmic changes without disrupting semantic content. These modified frame features are then reconstructed into perceptually natural audio waveforms by a high-fidelity vocoder ensuring the stealthiness of the embedded trigger.

Our contributions can be summarized as follows:

1) We pioneer a backdoor attack method operating purely in the latent space frame-level features, significantly enhancing attack stealthiness by avoiding direct, easily detected modifications to the audio.
2) We propose an efficient and flexible attack framework utilizing the capabilities of SSL models for robust feature extraction and neural vocoders for high-quality synthesis, also enabling partial feature transformations, like grapheme-level manipulation.
3) We conduct comprehensive experiments demonstrating that our latent-space backdoor attacks method achieves high attack success rates while remaining highly stealthiness from both human perception and automated analysis.

[1] School of Computer Science, XiangTan University, China
[2] School of Software & Microelectronics, Peking University, China
[*] Corresponding author. weipingwen@pku.edu.cn



(a) Poisoning Stage (b) Training Stage (c) Inference Stage

Fig. 1: Pipeline of the proposed framework LFBA.

II. BACKGROUND

A. Backdoor attacks

Backdoor attacks have been studied in the image and text classification domain [13], [14]. Existing studies have shown that similar perturbation-based techniques can also be used to generate triggers for effective backdoor attacks in speech models, such as noise clip [15], ultrasonic pulse [6] and environmental sound mimicking [7]. Recently, some researchers began to embed triggers by modifying the speech attributes, such as timbre conversion [10], emotion conversion [16], and rhythm alteration [11]. However, manipulating representations in latent space remains underexplored.

B. Self-Supervised Speech Representation Learning Models

Self-supervised learning (SSL) has significantly advanced speech processing by enabling models to learn rich, contextualized representations from large amounts of unlabeled audio data. These representations capture various acoustic and linguistic properties, proving highly effective for downstream tasks.

Our framework is agnostic to the specific feature extractor, allowing potentially any model that outputs frame-level representations. So we utilize WavLM [17] for frame-level feature extraction due to its demonstrated ability to capture detailed acoustic patterns and contextual information within its frame-level features. In addition, models like wav2vec2 [18] trained with Connectionist Temporal Classification (CTC) learn to map continuous speech signals to discrete units like graphemes and implicitly determine the corresponding time boundaries within the audio. So we utilize its ability of grapheme-boundary predictions to enable precise temporal segmentation of extracted features, bridging frame-level representations with linguistic units.

C. Neural Vocoders

Neural vocoders, such as HiFi-GAN [19], play a pivotal role in reconstructing high-fidelity waveforms from modified spectral or latent representations. A recent study [20] leverages it to reconstruct voice-converted speech from altered frame-level features while preserving speaker identity and content intelligibility. We are inspired to adopt HiFi-GAN to synthetic poisoned audio from manipulated WavLM features, ensuring perceptual naturalness and continuity.

III. METHODOLOGY

A. Preliminaries

Threat Model. In this work, we consider a third-party data poisoning attack scenario. We assume an adversary who does not have direct access to the user's model architecture or the training process at its initiation. However, the adversary possesses the capability to access and manipulate a portion of the training data used by the user. This manipulation includes the ability to alter the content of benign audio samples and to potentially fabricate corresponding labels. The user then unknowingly trains their model on the poisoned dataset.

Adversary's Goals. The primary objectives of the adversary are three-fold: effectiveness, stealthiness, and robustness. For effectiveness, the backdoored model should achieve a high attack success rate (ASR), i.e., misclassify triggered inputs to the target label. For Stealthiness, Poisoned samples must preserve naturalness in both auditory perception and machine-based metrics like speaker verification consistency, low word error rate. For Robustness, the attack should withstand common defenses and remain effective under real-world distortions.

Backdoor Attacks Pipeline The backdoor attacks pipeline consists of three stages as shown in Figure 1. In the attack stage, the adversary constructs a poisoned dataset $\mathcal{D}'$ (detailed in Section III-D). In the training stage, the victim model M_{cls} is trained on the poisoned dataset $\mathcal{D}'$. During training, M_{cls} learns two associations: (1) benign mappings between clean samples and their true labels, and (2) malicious mappings between poisoned samples (with embedded triggers) and the target label $\mathbf{y}_{\text{T}}$. This dual objective is achieved through standard cross-entropy optimization without requiring architecture modifications. In the inference stage, M_{cls} behaves normally on clean inputs $\mathbf{x}$ but misclassifies any poisoned input $\mathbf{x}'$ as $\mathbf{y}_{\text{T}}$.

B. Poisoned Inputs Generation

Step 1: Frame-Level Feature Extraction. For a given benign utterance $\mathbf{x}$, we first extract its frame-level feature representation. We pass the audio waveform through a WavLM model and obtain the output from a specific intermediate layer. This output is a sequence of feature vectors, where each vector corresponds to a short time frame of the audio. Let the resulting frame-level feature sequence of the benign utterance be

$$\mathcal{F} = \{f_1, f_2, ..., f_T\} \in \mathbb{R}^{T \times D} \ ,$$

where T is the total number of frames and D is the feature dimension.

Step 2: Grapheme-level Feature Alignment (Optional). We utilize a wav2vec2 model fine-tuned for CTC-based automatic speech recognition to obtain a grapheme-level transcription sequence $\mathcal{G}$ and the corresponding time intervals $\{\mathbf{I}_i\}_1^M$ where

$$\mathcal{G} = \{g_i\}_1^M = \{g_1, g_2, ..., g_M\} \ , \ \mathbf{I}_i = [t_{\text{start}}^{(i)}, t_{\text{end}}^{(i)}] \ .$$

So we can partition $\mathcal{F}$ into M disjoint subsequences $\{\mathcal{F}_i\}_1^M$ for each grapheme g_i in the utterance.

Step 3: Feature Transformation. Given a target grapheme subset $\mathcal{G}_{\text{target}}$ mapped to $\mathcal{F}_{\text{target}}$ where

$$\mathcal{G}_{\text{target}} = \{g_i\}_{i_{\text{start}}}^{i_{\text{end}}} \subseteq \mathcal{G} \ , \ \mathcal{F}_{\text{target}} = \textstyle\bigcup_{i_{\text{start}}}^{i_{\text{end}}} \mathcal{F}_i \subseteq \mathcal{F} \ .$$

We apply one of the four proposed feature transformation strategies to $\mathcal{F}_{\text{target}}$ (detailed in Section III-C), and get the modified frame-level feature sequence $\mathcal{F}'_{\text{target}}$. This can be configured in two ways: (1) Whole Utterance Transformation and (2) Grapheme-Specific Transformation identified using the alignment from Step 2. Therefore, the manipulated feature sequence $\mathcal{F}'$ can be simply expressed as:

$$\mathcal{F}' = \left(\mathcal{F} \setminus \mathcal{F}_{\text{target}}\right) \cup \mathcal{F}'_{\text{target}} \ .$$

Step 4: Waveform Reconstruction. Afterwards, the manipulated feature representation $\mathcal{F}'$ is fed into a HiFi-GAN vocoder to synthesize the corresponding audio sample $\mathbf{x}'$.

C. Frame-level Feature Transformation Strategies

We design four strategies centered on two dichotomies: compression versus expansion, similarity-guided versus simple transformations. Similarity-Guided Compression and Expansion use content-aware logic to preserve acoustic continuity, aiming for enhanced stealthiness. In contrast, Simple Compression and Expansion apply uniform, deterministic modifications, creating a consistent trigger pattern.

Similarity-Guided Feature Compression (SGC). To compress the target feature sequence $\mathcal{F}'_{\text{target}}$ by removing redundant or highly similar frame-level features, we compute adjacent pairwise cosine similarity $\mathcal{S}_{t-1,t}$ and $\mathcal{S}_{t,t+1}$ for f_t in $\mathcal{F}'_{\text{target}}$ (excluding the first and last frame-level feature) and remove adjacent frame-level features if they are both greater than preset threshold τ. To preserve spectral continuity, for consecutive eligible frame-level features $f_{\boldsymbol{t}}$ and $f_{\boldsymbol{t+1}}$, additional calculations of $\mathcal{S}_{\boldsymbol{t},t+2}$ and $\mathcal{S}_{t-1,\boldsymbol{t+1}}$ are required. We retain only the frame with the highest average similarity to its neighbors, as it exhibits greater contextual consistency, and remove the more redundant counterpart. For detailed implementation, see Algorithm 1.

Algorithm 1 Similarity-Guided Feature Compression (SGC)

```
Input: Target frame-level features $\mathcal{F}_{\text{target}}$, threshold $\tau$
Output: Compressed features $\mathcal{F}'_{\text{target}}$
if $T' \leq 3$ then
    return $\mathcal{F}_{\text{target}}$
end if
Compute pairwise similarities and get candidate indices
    $\mathcal{C} = \left\{ i \,\middle|\, 2 \leq i \leq T'-1, \ \mathcal{S}_{i,i+1} = \cos(f_i, f_{i+1}) \geq \tau \right\}$
Initialize removal set $\mathcal{R} \leftarrow \emptyset$
for each consecutive pair $(i, i+1)$ in $\mathcal{C}$ do
    Calculate average similarities
    $\bar{\mathcal{S}}_i = (\mathcal{S}_{i-1,i} + \mathcal{S}_{i,i+1} + \mathcal{S}_{i,i+2})/3, \ \bar{\mathcal{S}}_{i+1} = (\mathcal{S}_{i,i+1} + \mathcal{S}_{i+1,i+2} + \mathcal{S}_{i-1,i+1})/3$
    if $\bar{\mathcal{S}}_i \geq \bar{\mathcal{S}}_{i+1}$ then
        $\mathcal{R} \leftarrow \mathcal{R} \cup \{i-1, i+1\}$
    else
        $\mathcal{R} \leftarrow \mathcal{R} \cup \{i, i+2\}$
    end if
end for
for each isolated index $i$ in $\mathcal{C}$ do
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{i-1, i+1\}$
end for
Construct compressed sequence
    $\mathcal{F}'_{\text{target}} = \mathcal{F}_{\text{target}} \setminus \{f_j \in \mathcal{F}_{\text{target}} \,|\, j \in \mathcal{R}\}$
return $\mathcal{F}'_{\text{target}}$
```

Similarity-Guided Feature Expansion (SGE). Likewise, we insert an averaged feature between adjacent frame-level features f_t and f_{t+1} if their cosine similarity $\mathcal{S}_{t,t+1}$ exceeds τ to ensure smooth transitions, mimicking natural speech rate variations.

Simple Feature Compression (SC). As a comparison, we propose a straightforward way to compress the target feature sequence $\mathcal{F}'_{\text{target}}$ by removing every other frame-level feature. To be specific, delete odd-indexed features within $\mathcal{F}'_{\text{target}}$, approximately halving the temporal resolution.

Simple Feature Expansion (SE). As a comparison, we propose a straightforward way to expand the target feature sequence $\mathcal{F}'_{\text{target}}$ by duplicating each frame-level feature f_t.

D. Poisoned Dataset Generation

We introduce a backdoor into speech classification models via the poisoned-label attack. Given a benign dataset $\mathcal{D}$, the attacker specify target subset

$$\mathcal{D}_t = \{(\mathbf{x}_i, y_i)\}_1^n \subset \mathcal{D} \ , \quad n < N$$

and apply the trigger function $F_t(\cdot)$ to each utterance $\mathbf{x}_i \in \mathcal{D}_t$. We also associate these triggered samples with the

TABLE I: Attack results on GSC dataset towards KWS task. Each item shows evaluations AV (%) / ASR (%) in the table.

	Resnet-34	Attention-LSTM	KWS-ViT	EAT-S
DABA	1.34 / 99.13	1.21 / 98.89	1.47 / 99.02	1.95 / 98.45
Ultrasonic	0.73 / 99.22	1.15 / 98.93	0.96 / 98.76	1.02 / 98.94
VSVC	0.44 / 99.20	0.53 / 98.87	0.65 / 99.43	0.48 / 98.63
RSRT(Stretch)	0.65 / 99.30	0.59 / 99.24	0.42 / 99.01	0.67 / 98.41
RSRT(Squeeze)	0.41 / 99.05	0.52 / 99.32	0.69 / 99.25	0.73 / 98.79
LFBA(SGC)	0.51 / 99.56	0.63 / 99.39	0.82 / 99.42	0.86 / 98.81
LFBA(SC)	0.38 / 99.61	0.44 / 99.23	0.52 / 99.66	0.78 / 99.10
LFBA(SGE)	0.35 / 99.52	0.39 / 99.09	0.61 / 99.60	0.72 / 98.93
LFBA(SE)	0.29 / 99.74	0.40 / 99.19	0.58 / 99.61	0.67 / 99.15

TABLE II: The stealthiness evaluation for different backdoor attack methods.

	w/o trigger	DABA	Ultrasonic	VSVC	RSRT (Stretch)	RSRT (Squeeze)	**LFBA (SGC)**	**LFBA (SC)**	**LFBA (SGE)**	**LFBA (SE)**
$NISQA$	3.34	2.83	2.47	3.40	3.03	2.68	3.79	2.87	3.15	3.27
MOS	3.50	3.16	2.88	3.53	3.21	3.19	3.72	3.24	3.65	3.66
TCR(%)	99.3	61.2	79.6	10.1	92.9	87.4	97.2	94.1	98.9	98.7

attacker-specified label $\mathbf{y}_\mathrm{T}$ to replace y_i. So the final poisoned dataset is constructed as

$$\mathcal{D}' = \mathcal{D} \setminus \mathcal{D}_t \cup \mathcal{D}'_t \text{ , where } \mathcal{D}'_t = \{(F_t(\mathbf{x}_i), \mathbf{y}_\mathrm{T})\}_1^n \ .$$

IV. Experiments and Results

A. Main Settings

Dataset. We evaluate our method on Google Speech Commands (GSC) [4], a widely used benchmark for KWS research. The dataset contains 64,721 one-second audio clips across 30 classes, recorded by thousands of speakers in diverse acoustic conditions. Following prior work [11], we select 10 keywords for evaluation: "stop", "go", "yes", "no", "up", "down", "left", "right", "on", "off" .

Victim models. We select several classic classification models used in classification tasks as our attack targets. (1) ResNet34 [21], which is a classic classification model from the early days of speech recognition tasks. (2) Attention-LSTM [22], which adds an attention mechanism, enabling sequence modeling capabilities. (3) KWS-VIT [23], which combines transformer architecture. (4) EAT-S [24], which is a typical end-to-end model that combines CNNs.

Baseline. We compare our attack with some representative speech backdoor attacks: (1) Dual-Adaptive Backdoor Attacks (DABA) [15], (2) Ultrasonic voice as the trigger (Ultrasonic) [6], (3) Voiceprint Selection and Voice Conversion (VSVC) [10] and (4) Random Spectrogram Rhythm Transformation (RSRT) [11].

Attack Setup. We set the poisoning rate γ to 1%, indicating the proportion of samples poisoned by the trigger in training data. The target label $\mathbf{y}_\mathrm{T}$ for the backdoor attacks method is set to "yes" to simulate a high-risk attack scenario. Frame-level features are extracted from the 6th layer of the WavLM-Large model with dimension $D = 1024$. The similarity threshold τ used in the feature transformation strategies is set to 0.75. We employed the whole utterance transformation configuration here, applying the transformations to the entire frame-level feature sequence of each sample. This choice was based on preliminary analyses and ablation studies, which indicated this approach yielded a superior trade-off between attack success rate and stealthiness for our main comparisons. Speech synthesis is performed using a HiFi-GAN vocoder, following the setup in the KNN-VC work [20]. For all baselines, follow their original configurations if any settings are not explicitly mentioned here.

Training Setup. We trained all the victim models with the same hyperparameters. The batch size is 64. The weights are optimized by Adam optimizer with a learning rate of 1e-4 and cross-entropy loss function. We trained 30 epochs to make all models converge. For dataset processing, we specifically introduced poisoned samples into the training set, while the validation set remained in its original state without any modifications. All experiments were conducted using the PyTorch framework on a Nvidia RTX 4090 GPU.

B. Evaluation Metrics

For effectiveness, we examine the Accuracy Variance (AV) and the Attack Success Rate (ASR) for each attack. The AV represents the model's accuracy change after the trigger is applied during training. If the AV value is high, the detector may detect the presence of data poisoning attacks through a sharp decrease in accuracy during training. The ASR stands for the hit rate of the trigger on the test set.

For stealthiness, we evaluate the attack's stealthiness from the following perspectives: (1) Objective Audio Quality: We use the $NISQA$ score to measure the objective quality of the synthesized audio. $NISQA$ [25] is a non-intrusive metric that predicts the perceptual quality of speech. A higher $NISQA$ score indicates better audio quality. (2) Subjective Perceptual Evaluation: We conduct a Mean Opinion Score (MOS) test to assess the subjective perceptual quality of the synthesized audio. 15 participants rated the perceptual quality of 20 poisoned audio samples respectively on a scale of 1 (worst) to 5 (best). (3) Speech Timbre Consistency: We employ a speaker verification model ERes2Net [26] to measure the consistency of the speaker's voice before and after the backdoor trigger is applied. The Timbre Consistency Rate (TCR) represents the similarity between the speaker embeddings extracted from the original and synthesized audio. A higher consistency score indicates better preservation of the speaker's timbre.

TABLE III: The performance of different $\mathcal{F}_{\text{target}}$ using SGC and SE strategies on ResNet34.

	w/o trigger	vowel (SGC)	vowel (SE)	consonant (SGC)	consonant (SE)	**whole** **(SGC)**	**whole** **(SE)**
ASR (%)	\	99.10	99.36	98.88	98.72	99.56	99.74
AV (%)	\	1.09	0.84	2.67	1.84	0.51	0.29
$NISQA$	3.34	3.26	3.35	3.52	3.39	3.79	3.27
MOS	3.50	3.82	3.73	3.88	3.81	3.72	3.66
TCR (%)	\	98.2	98.8	98.4	99.1	97.2	98.7

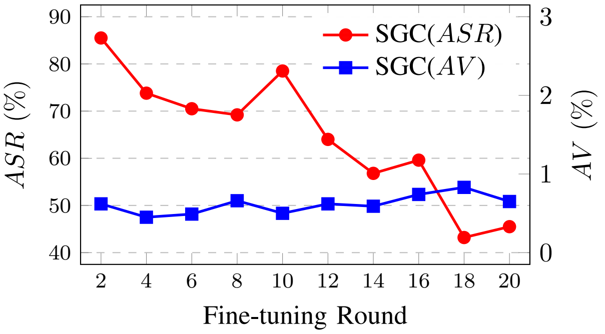


Fig. 2: The resistance to fine-tuning.

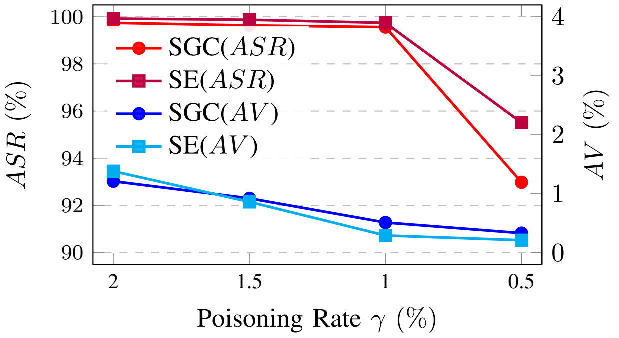


Fig. 3: The performance of different poisoning rate γ.

For robustness, we evaluate the robustness of our backdoor attacks method by measuring its Resistance to Fine-tuning [27]. We fine-tune the backdoored model on a benign dataset for 20 epochs. A robust backdoor attack method should still maintain a high ASR.

C. Main Result

Attack Effectiveness. Table I demonstrates the effectiveness of our proposed LFBA variants compared to baseline methods across four KWS architectures. All strategies achieve excellent average ASR with minimal AV, confirming the viability of manipulating frame-level latent features for potent backdoor injection. Notably, the simple strategies (SC/SE) exhibit slightly higher ASR compared to their similarity-guided counterparts (SGC/SGE), suggesting that their uniform transformations likely create a more consistent and easily distinguishable pattern for the model to learn during training.

Attack Stealthiness. As shown in Table II, LFBA achieves the highest $NISQA$ score, indicating near-benign audio quality, while Ultrasonic suffer severe degradation. A similar trend is observed in MOS. LFBA also yields a higher TCR, confirming that latent-space manipulation better preserves speaker identity, whereas VSVC underperforming due to its explicit timbre conversion. Moreover, SGC outperforms SC, validating the effectiveness of our tailored strategy for stealthiness. SGE, however, slightly lags behind SE, likely due to the added perturbation from introducing new features versus amplifying existing ones.

Attack Robustness. Figure 2 reveals that ASR gradually declines from 99% to 45% after 20 epochs of fine-tuning with a benign dataset on the victim model ResNet34. However, the retained ASR remains significantly higher than 40%, indicating persistent backdoor activation. The fluctuating AV suggests partial but incomplete defense via catastrophic forgetting.

V. ABLATION STUDY

Chosen of strategies: As shown in Table I, compression-based strategies generally achieve higher ASR than expansion-based ones, with SGC strategy exhibiting superior stealthiness. This aligns with our hypothesis that selectively removing high-similarity frames preserves semantic continuity while introducing subtle rhythmic distortions. In contrast, SGC strategy marginally underperforms SE strategy in ASR, likely due to its deterministic duplication avoiding interpolation artifacts. The results suggest that redundancy-aware compression optimizes the trade-off between attack potency and imperceptibility.

Types of $\mathcal{F}_{\text{target}}$: While our main experiments applied transformations to the entire utterance's latent features ($\mathcal{F}_{\text{target}} = \mathcal{F}$) for optimal overall performance, our framework's design allows for more granular attacks by targeting specific linguistic units. To investigate the effect of this granularity, we leverage the grapheme-level alignment capability of our framework (Sec III-B, Step 2) to isolate and modify features corresponding only to vowels or consonants.

For this ablation, we selected representative keywords such as "$stop$", "yes", "up", and "$right$" and identified their core vowel and consonant graphemes. We applied two distinct transformation strategies, SGC and SE to different $\mathcal{F}_{\text{target}}$ and evaluated the results on the ResNet34 victim model. The performance comparison is presented in Table III.

Modifying entire segments yields the highest $NISQA$, outperforming vowel-only and consonant-only backdoor attacks. This is attributed to the holistic preservation of coarticulation patterns—critical for naturalness. Notably, vowel-focused attacks achieve higher ASR than consonant-based ones, as vowels' longer duration and spectral stability enhance latent-trigger consistency.

The investigation into target feature granularity demonstrates the flexibility of our LFBA framework. While targeting specific linguistic units like vowels or consonants is feasible and achieves high ASR, modifying the whole utterance provides the most potent attack in terms of ASR and AV. Stealthiness metrics result present a perceptual divergence: human listeners prioritize rhythmic continuity (favoring grapheme-specific transformation), whereas $NISQA$ emphasizes spectral fidelity (favoring whole utterance transformation). Modifying entire segments maintains the natural spectral transitions between phonemes, which is critical for speaker consistency.

Overall, these findings validate our use of the whole utterance transformation for the main experimental evaluation while highlighting the potential for more targeted attacks depending on the adversary's specific goals and constraints.

Effects on poisoning rate γ: Figure 3 reveals that ASR remains robust on victim model ResNet34 even at $\gamma = 0.5\%$, with SE strategy outperforming SGC strategy (95.51% for SE and 92.98% for SGC). The marginal ASR drop at low γ suggests latent-space triggers exhibit high memorability despite sparse poisoning. Concurrently, AV decreases with γ from 1.38 to 0.21 for SE, indicating minimal interference with benign task learning. This confirms LFBA's viability in low-resource attack scenarios without sacrificing stealthiness.

VI. Conclusions

This paper introduces LFBA, a latent-space frame-level backdoor attack framework. LFBA embeds triggers through stealthy temporal dynamics alterations without semantic content or speaker characteristics changes. Extensive experiments validate LFBA's effectiveness, stealthiness, and robustness. Compared to prior methods, LFBA eliminates reliance on perceptible signal modifications. Our work further demonstrated the flexibility with more granular attacks on targeting specific linguistic units except for an entire utterance. This adaptability, coupled with the core mechanism of latent-space frame-level manipulation, suggests that the principles behind LFBA are not limited to KWS. We believe this approach has broader implications, potentially serving as a blueprint for stealthy backdoor attacks or highlighting vulnerabilities in a wider array of speech processing applications. Consequently, our findings underscore the critical need for developing robust defenses against such latent-space threats and for promoting secure data handling practices across all speech and audio-related domains.

References

[1] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 4087–4091, IEEE, 2014.

[2] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting.," in *Interspeech*, pp. 1478–1482, 2015.

[3] P. Warden, "Speech commands: a public dataset for single-word speech recognition (2017)," *Dataset available from http://download. tensorflow. org/data/speech_commands_v0*, vol. 1, 2017.

[4] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[5] T. Zhai, Y. Li, Z. Zhang, B. Wu, Y. Jiang, and S.-T. Xia, "Backdoor attack against speaker verification," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2560–2564, IEEE, 2021.

[6] S. Koffas, J. Xu, M. Conti, and S. Picek, "Can you hear it? backdoor attacks via ultrasonic triggers," in *Proceedings of the 2022 ACM workshop on wireless security and machine learning*, pp. 57–62, 2022.

[7] C. Shi, T. Zhang, Z. Li, H. Phan, T. Zhao, Y. Wang, J. Liu, B. Yuan, and Y. Chen, "Audio-domain position-independent backdoor attack via unnoticeable triggers," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, pp. 583–595, 2022.

[8] P. Liu, S. Zhang, C. Yao, W. Ye, and X. Li, "Backdoor attacks against deep neural networks by personalized audio steganography," in *2022 26th International Conference on Pattern Recognition (ICPR)*, pp. 68–74, IEEE, 2022.

[9] S. Koffas, L. Pajola, S. Picek, and M. Conti, "Going in style: Audio backdoors through stylistic transformations," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, IEEE, 2023.

[10] H. Cai, P. Zhang, H. Dong, Y. Xiao, S. Koffas, and Y. Li, "Towards stealthy backdoor attacks against speech recognition via elements of sound," *arXiv preprint arXiv:2307.08208*, 2023.

[11] W. Yao, J. Yang, Y. He, J. Liu, and W. Wen, "Imperceptible rhythm backdoor attacks: Exploring rhythm transformation for embedding undetectable vulnerabilities on speech recognition," *Neurocomputing*, vol. 614, p. 128779, 2025.

[12] B. Xiong, Z. Xing, and W. Wen, "Phoneme substitution: A novel approach for backdoor attacks on speech recognition systems," in *2024 IEEE 36th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 540–547, IEEE, 2024.

[13] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47230–47244, 2019.

[14] X. Pan, M. Zhang, B. Sheng, J. Zhu, and M. Yang, "Hidden trigger backdoor attack on {NLP} models via linguistic style manipulation," in *31st USENIX Security Symposium (USENIX Security 22)*, pp. 3611–3628, 2022.

[15] Q. Liu, T. Zhou, Z. Cai, and Y. Tang, "Opportunistic backdoor attacks: Exploring human-imperceptible vulnerabilities on speech recognition systems," in *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 2390–2398, 2022.

[16] W. Yao, Z. X. Chen, J. Liu, W. Wen, *et al.*, "Emoattack: Utilizing emotional voice conversion for speech backdoor attacks on deep speech classification models," *arXiv preprint arXiv:2408.15508*, 2024.

[17] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[18] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12449–12460, 2020.

[19] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17022–17033, 2020.

[20] M. Baas, B. van Niekerk, and H. Kamper, "Voice conversion with just nearest neighbors," in *Interspeech 2023*, pp. 2053–2057, 2023.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[22] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," *arXiv preprint arXiv:1704.02971*, 2017.

[23] A. Berg, M. O'Connor, and M. T. Cruz, "Keyword transformer: A self-attention model for keyword spotting," *arXiv preprint arXiv:2104.00769*, 2021.

[24] A. Gazneli, G. Zimerman, T. Ridnik, G. Sharir, and A. Noy, "End-to-end audio strikes back: Boosting augmentations towards an efficient audio classification network," *arXiv preprint arXiv:2204.11479*, 2022.

[25] G. Mittag, B. Naderi, A. Chehadi, and S. Möller, "Nisqa: A deep cnn-self-attention model for multidimensional speech quality prediction with crowdsourced datasets," in *Proc. Interspeech 2021*, pp. 2127–2131, 2021.

[26] Y. Chen, S. Zheng, H. Wang, L. Cheng, Q. Chen, and J. Qi, "An enhanced res2net with local and global feature fusion for speaker verification," in *Interspeech 2023*, pp. 2228–2232, 2023.

[27] Y. Liu, Y. Xie, and A. Srivastava, "Neural trojans," in *2017 IEEE International Conference on Computer Design (ICCD)*, pp. 45–48, IEEE, 2017.