REVISITING VOCAL-TRACT-BASED DETECTION FOR MODERN AUDIO DEEPFAKE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vocal-tract-based audio deepfake detection relies on the premise that deepfake generators may fail to reproduce the vocal-tract characteristics of natural human speech. However, its robustness against modern synthesis models and adversarial manipulation remains insufficiently understood. We systematically evaluate vocal-tract detection against recent text-to-speech (TTS) and voice conversion (VC) models and show that its effectiveness degrades substantially on modern models. Through ablation studies, we identify two major contributors to vocal-tract realism: learned linguistic representations in TTS models and explicit modeling of source-speech fundamental frequency in VC models. We further introduce *VoicePatch*, a model-agnostic post-generation editing attack that evades detection by replacing detector-sensitive phoneme segments with real speech. Across three evaluated deepfake models, VoicePatch increases detection bypass rates from below 2% to above 80%.

1 INTRODUCTION

Concerned about the growing misuse of deep learning-based speech synthesis Stupp; Tian et al. (2020), commonly referred to as audio deepfakes, researchers have devoted substantial effort to developing effective detection methods. Many existing studies use neural networks to identify acoustic artifacts or statistical patterns in synthesized speech Subramani & Rao (2020); Alzantot et al. (2019). Although these methods can achieve strong performance against known generation models, their effectiveness often degrades when confronted with unseen synthesis techniques Subramani & Rao (2020). A separate line of work takes a more principled approach by examining whether speech conforms to the physiological and acoustic constraints of human speech production Mostaani & Magimai Doss (2022). Vocal-tract-based detection is a representative approach in this direction Blue et al. (2022). It approximates the vocal tract as a sequence of acoustic tubes and estimates parameters such as tube cross-sectional areas or reflection coefficients from speech signals. Deepfake models that fail to faithfully reproduce the human speech-production process may yield abnormal vocal-tract estimates that distinguish synthetic from natural speech.

Despite its strong physical motivation and promising performance against earlier deepfake techniques, vocal-tract-based detection faces two emerging challenges. First, compared with earlier synthesis systems, modern deepfake models are trained on substantially larger speech datasets and employ more capable architectures Wang et al. (2023); Shen et al. (2024). These advances allow them to better approximate the distribution of natural speech across a wide range of acoustic and statistical dimensions, potentially including vocal-tract characteristics. Second, although incorporating detector-evasion objectives directly into an end-to-end synthesis pipeline can be computationally expensive and impractical Blue et al. (2022), an attacker may instead manipulate synthetic speech after generation to evade vocal-tract detection. Whether such post-generation manipulation can effectively bypass detection remains largely unexplored.

To systematically assess the robustness of vocal-tract-based detection, we revisit this approach through an extensive experimental study and investigate post-generation editing. We first evaluate vocal-tract detection against a diverse set of modern text-to-speech (TTS) and voice conversion (VC) models and show that its effectiveness degrades substantially on recent deepfake models. We further conduct ablation studies to identify the model components responsible for increased vocal-tract realism. Our results show that learned linguistic representations in TTS models and explicit

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

modeling of the source speech's fundamental frequency in VC models are major contributors. These findings provide guidance for both future detection methods and potential defenses that deliberately limit the vocal-tract realism reproduced by generation models. Finally, we demonstrate the feasibility of post-generation evasion through a new attack, *VoicePatch*. VoicePatch replaces phoneme segments targeted by the detector with corresponding clips from real speech and uses a lightweight smoothing method to seamlessly blend the synthetic and real waveforms around each splice. Experiments show that VoicePatch achieves an attack success rate of over 80% while introducing only minor degradation in speech quality.

We organize this paper as follows. § 2 introduces the background. § 3 and § 4 report the evaluation and ablation studies on diverse TTS and VC models, respectively. § 5 presents the VoicePatch design and evaluates its performance. § 6 discusses related work, and § 7 concludes the paper.

2 BACKGROUND

Vocal-Tract-Based Detection. Vocal-tract-based detection identifies deepfake audios by examining whether the speech conforms to the physiological constraints of the human vocal tract. During human speech production, sound generated by vocal-fold vibration propagates through the vocal tract and is shaped by reflections caused by its varying cross-sectional area. To characterize this process, detection approaches typically approximate the vocal tract as a cascade of equal-length, lossless tube sections with different cross-sectional areas. At the junction between the k-th and $k+1$-th sections, the acoustic wave is reflected with a coefficient given by $r_k = \frac{A_k - A_{k+1}}{A_k + A_{k+1}}$, where A_k and A_{k+1} denote the cross-sectional areas of the two sections. Several algorithms have been developed to estimate these reflection coefficients and corresponding tube areas from speech signals Rabiner & Schafer (1978). For genuine human speech, the recovered tube-area profiles are consistent with the physiological constraints of the human vocal tract. In contrast, synthetic speech generated without faithfully reproducing vocal-tract acoustics may produce abnormal tube-area estimates.

Modern Audio Deepfake Techniques. Audio deepfake techniques synthesize or manipulate speech to convey intended linguistic content while imitating a target speaker's voice. They can generally be divided into text-to-speech (TTS) synthesis and voice conversion (VC) Khanjani et al. (2023); Li et al. (2025a). TTS models synthesizes speech from textual input and typically consists of a text encoder, a speaker encoder, and a speech-generation module Jia et al. (2018). The text encoder transforms the input text into linguistic representations, while the speaker encoder extracts a speaker embedding from one or more reference utterances to capture the target speaker's vocal characteristics. In this work, we use the term *vocoder* broadly to refer to the speech-generation module that converts the linguistic and speaker representations into an output waveform. Depending on the architecture, the vocoder may generate the waveform directly using an end-to-end neural network Kim et al. (2021), or first predict an intermediate acoustic representation, such as a mel-spectrogram, which is then converted into speech by a waveform generator Shen et al. (2018). VC models transform speech from a source speaker while preserving its linguistic content and reproducing the target speaker's voice. A typical VC pipeline consists of a content encoder, a speaker encoder, and a vocoder Sisman et al. (2020); Li et al. (2023a). The content encoder extracts a linguistic representation from the source speech. Some models additionally extract prosodic features, such as fundamental frequency, and integrate them with the linguistic representation. The speaker encoder extracts the target speaker's vocal characteristics from one or more reference utterances. The vocoder then combines the source-content representation, target-speaker embedding, and, when applicable, prosodic features to generate the converted speech. As with TTS, this generation may be performed end to end or through an acoustic decoder followed by a waveform generator.

3 VOCAL-TRACT REALISM IN MODERN TTS MODELS

To evaluate the vocal-tract realism of TTS models, we implement the detector proposed in Blue et al. (2022). The detector models the vocal tract as $N = 2LF/C$ tube sections, where L, F, and C denote the vocal-tract length, audio sampling rate, and speed of sound, respectively. We set L to the average human vocal-tract length of $15.5\,\mathrm{cm}$. The detector estimates and examines tube cross-sectional areas from audio segments that each contains two adjacent phonemes, referred to as bigrams. Following the original design, each bigram is divided into $35\,\mathrm{ms}$ frames with an overlap of $7\,\mathrm{ms}$. For

each frame, the detector iteratively adjusts the reflection coefficients $(r_0, \ldots, r_{N-1})$ until the resulting spectrum sufficiently matches the observed audio spectrum. By setting the cross-sectional area at the glottis to the average human value of $3.7\,\mathrm{cm}^2$, the detector then converts the estimated reflection coefficients into tube-area estimates. The detector profiles a large number of instances of each bigram type to characterize the normal ranges of tube-area features in real speech. According to Blue et al. (2022), only certain tube-section areas associated with particular bigram types can reliably distinguish synthetic from real speech. We therefore adopt the same feature-selection procedure. To validate our implementation, we evaluate the detector under the same setting as Blue et al. (2022), using TIMIT Garofolo et al. (1993) as the real-speech dataset and Real-Time Voice Cloning (RTVC) Jemine (2019) as the synthetic model. Our implementation achieves comparable detection performance across multiple metrics, confirming its consistency with the original detector. Please see App. A for additional details on detector implementation and evaluation. Notably, RTVC is a TTS model. To avoid confusion, we refer to it as RT Voice Cloning hereafter.

3.1 TTS Model Selection and Detection Performance

Among existing TTS deepfake models, we focus on zero-shot models, which require only a few reference utterances from the target speaker and are therefore particularly practical for impersonation attacks Azzuni & El Saddik (2025). Prior studies suggest that larger models and higher-dimensional internal representations can improve speech synthesis performance Niu et al. (2025), so we intentionally select models spanning a broad range of model sizes and linguistic embedding dimensions, including YourTTS Casanova et al. (2022), StyleTTS2 Li et al. (2023b), and XTTS Casanova et al. (2024). Please see App. B for the configuration details of those models. We evaluate each model over five experimental sessions. In each session, we construct a detector using a different set of speakers for feature selection. Tab. 1 reports the average detection results and includes RT Voice Cloning for comparison. Although none of the three modern models explicitly incorporates vocal-tract modeling into its training or speech generation, the detector's performance degrades substantially across all three. While individual metrics vary, the overall accuracy remains below 64%, and the precision is consistently low. A recall above 80% may appear promising, but it is accompanied by an FPR as high as 59.80%, making the detector impractical. Overall, these results indicate that vocal-tract-based detection is no longer effective against modern TTS deepfake models.

Table 1: Detection performances on TTS models.

Model	Accuracy	Precision	Recall	FPR
RT Voice Cloning	99.48%	99.49%	98.84%	0.51%
YourTTS	60.32%	59.60%	55.70%	35.40%
StyleTTS2	63.54%	64.49%	81.29%	59.80%
XTTS	52.35%	54.01%	57.60%	53.40%

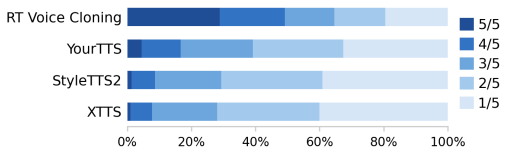

Figure 1: Bigram selection statistics of TTS models.

We further evaluate the stability of feature selection across experimental sessions. For each TTS model, we aggregate the bigram types selected over the five sessions. For RT Voice Cloning, an average of 309 bigram types are selected per session, with 478 unique types selected across all five sessions. The corresponding numbers are 274 and 602 for YourTTS, 207 and 519 for StyleTTS2, and 210 and 534 for XTTS. Compared with RT Voice Cloning, the three modern models yield fewer selected bigram types per session but substantially more unique types across sessions, indicating lower selection stability across different speaker sets. To quantify this difference, we count the number of sessions in which each bigram type is selected and report the distribution in Fig. 1. For RT Voice Cloning, more than half of the selected bigram types appear in at least four sessions, and 28.87% are selected in all five sessions, indicating that a substantial portion provide stable, speaker-generalizable discriminative features. In contrast, the selection stability decreases considerably for the three modern models. For YourTTS, only 16.61% of the selected bigram types appear in at least four sessions, and 4.49% appear in all five. These proportions further decrease to 8.67% and 1.35% for StyleTTS2, and to 7.68% and 0.94% for XTTS, respectively. These results suggest that the detector struggles to identify stable and generalizable bigram features for modern TTS models. This trend is consistent with the results in Tab. 1: models with less stable bigram selection are more difficult to detect.

We also find that, although the modern models yield a larger set of unique selected bigram types across sessions, occurrences of these bigrams collectively cover a much smaller portion of the evaluated utterances than those selected for RT Voice Cloning. On average, the speech segments cor-

responding to selected bigrams cover 64.91% of the total utterance duration for RT Voice Cloning, compared with only 34.74%, 34.32%, and 32.45% for YourTTS, StyleTTS2, and XTTS, respectively. This observation suggests that the detectors for modern TTS models increasingly rely on relatively infrequent bigram types whose limited samples in the feature-selection dataset may happen to exhibit distinguishable characteristics, rather than on bigrams that provide broadly applicable discriminative features. Consequently, the selected features generalize poorly across utterances and speaker sets.

3.2 Ablation Study on TTS Vocal-Tract Realism

The preceding findings motivate us to investigate which components of modern TTS architectures contribute to vocal-tract realism. To this end, we conduct a series of ablation experiments that selectively suppress or emphasize individual modules to isolate their contributions.

Metric Design. We introduce a metric based on vocal-tract estimates to quantify the similarity between synthetic and real speech. Specifically, we compare audio frames belonging to the same phoneme category and measure the proportion of synthetic–real frame pairs whose vocal-tract estimates exhibit a Pearson correlation exceeding a threshold τ_s. Since vowels exhibit strong periodic excitation and relatively stable articulation, they allow more reliable vocal-tract estimation Blue et al. (2022). We therefore restrict the measurement to vowel phonemes only, which does not harm our objective of measuring vocal-tract similarity. We refer to the metric as *vocal-tract fidelity*.

For each deepfake model, we generate synthetic utterances and combine them with real utterances from TIMIT to construct an evaluation dataset. Following the detector configuration, we divide each waveform into $35\,\mathrm{ms}$ frames and identify their phoneme labels using a fine-tuned wav2vec 2.0 model Baevski et al. (2020). Consecutive frames with the same predicted phoneme label are grouped into candidate phoneme intervals. We retain intervals containing at least two frames with an average classification confidence above 0.5. From each valid vowel interval, we select the two central frames to minimize the influence of articulatory transitions and obtain more reliable vocal-tract estimates. We then compute the Pearson correlation between the estimated tube-area vectors of synthetic and real frames belonging to the same vowel category. Vocal-tract fidelity is defined as the proportion of these frame pairs whose correlations exceed τ_s. We determine τ_s using real speech from TIMIT. Applying the same frame-selection and vocal-tract estimation procedures, we compute the Pearson correlations between the tube-area vectors of all pairs of real frames belonging to the same vowel category. We then set τ_s to the fifth percentile of the resulting correlation distribution.

Ablation of TTS Model Components. To identify the components responsible for vocal-tract realism in TTS models, we perturb their intermediate representations and measure the vocal-tract fidelity of the resulting speech. We represent the synthesis process as $\mathcal{A} = \mathcal{V}(R_t, R_s)$, where R_t and R_s denote the linguistic embedding sequence and speaker embedding, respectively, and $\mathcal{V}$ denotes the vocoder. To suppress speaker-specific information, we replace R_s with the average embedding of 1,000 real utterances, 10 from each of 100 speakers. This neutralization reduces speaker-specific information, including potential vocal-tract cues, while preserving a valid input to the vocoder.

We develop three methods for generating randomized linguistic embeddings that preserve different statistical properties of the original representations. To characterize these properties, we collect linguistic embeddings from 10,000 synthesis instances. Aggregating the m-th elements of all the linguistic representation vectors, we acquire an empirical set $\mathcal{D}_m = \{v_i^{n,m} \mid i = 1, \ldots, 10000,\ n = 1, \ldots, L_i\}$ for this feature dimension calculate its mean μ_m and standard deviation σ_m. Here, L_i is the length of the linguistic embedding sequences in the i-th synthesis instance. Based on these statistics, we develop the following three methods. All three generate sequences of length L, set to the maximum sequence length observed among the 10,000 synthesis utterances.

Parameter-guided (P). This method preserves the empirical mean and variance of each feature dimension while assuming a Gaussian distribution. Each embedding vector is generated independently, with its m-th element sampled from $\mathcal{N}(\mu_m, \sigma_m^2)$.

Distribution-guided (D). This method more closely approximates the empirical marginal distribution of each feature dimension. Specifically, the m-th element of the n-th embedding vector is generated as $\hat{v}^{n,m} = d^{n,m} + \alpha^{n,m}\sigma_m$, where $d^{n,m}$ is sampled uniformly from $\mathcal{D}_m$ and $\alpha^{n,m} \sim \mathcal{U}(-0.5, 0.5)$. All representation vectors are generated independently.

Dependency- and distribution-guided (DD). This method additionally approximates temporal dependencies between consecutive embedding vectors. Since modern text encoders contextualize each embedding using surrounding linguistic content, adjacent vectors are generally correlated. To capture these dependencies, we generate the first vector $\tilde{V}^1$ using the distribution-guided method and construct subsequent vectors recursively as $\tilde{v}^{n,m} = \beta_m \tilde{v}^{n-1,m} + \sqrt{1-\beta_m^2}, \tilde{w}^{n,m}$, where $\tilde{w}^{n,m}$ is independently generated using the distribution-guided method. The coefficient $\beta_m \in [0, 1]$ controls the dependency between consecutive vectors along feature dimension m and is estimated from the collected linguistic embeddings.

Table 2: Vocal-tract fidelity across eight settings (%).

Model	$L_O S_O$	$L_P S_O$	$L_D S_O$	$L_{DD} S_O$	$L_O S_N$	$L_P S_N$	$L_D S_N$	$L_{DD} S_N$
YourTTS	93.15	66.03	79.05	92.24	92.54	63.65	80.27	92.17
StyleTTS2	95.06	73.89	86.95	88.88	95.28	70.38	87.96	90.83
XTTS	94.64	71.82	89.32	91.68	94.65	69.29	89.46	92.13

Combining the four linguistic representation settings (original and the three randomized variants) with the two speaker embedding settings (original and neutralized) yields eight configurations. For each TTS model, we generate 20,000 utterances per configuration and measure their vocal-tract fidelity. Tab. 2 summarizes the results. Each configuration is denoted by a pair of labels: L_O, L_P, L_D, and L_{DD} represent the original, parameter-guided, distribution-guided, and dependency- and distribution-guided linguistic embeddings, respectively, while S_O and S_N indicate the original and neutralized speaker embeddings. For reference, we also evaluate unperturbed RT Voice Cloning, which achieves a vocal-tract fidelity of 65.94%.

The results show that speaker embedding neutralization has little effect on vocal-tract fidelity, causing negligible or no degradation. The vocoder alone also appears insufficient to reproduce the high vocal-tract realism of modern TTS models. Specifically, when supplied with parameter-guided linguistic embeddings and original speaker embeddings, the vocoder still generates speech containing a considerable number of recognizable phoneme segments. However, vocal-tract fidelity drops substantially, approaching that of RTVC in some cases. These results identify linguistic representations as a major contributor to vocal-tract realism. Both feature-wise distributions and temporal dependencies contribute to higher fidelity. Compared with parameter-guided randomization, preserving the empirical feature-wise distributions improves vocal-tract fidelity by more than 10 percentage points across all evaluated settings. Incorporating temporal dependencies yields a further improvement of 3–12 percentage points.

Takeaway. *Vocal-tract detection is ineffective against modern TTS models. Our ablation results further show that learned linguistic representations contribute substantially to vocal-tract realism. Future efforts to improve detection or limit the vocal-tract realism of synthetic speech should therefore pay particular attention to the feature-wise distribution and temporal dependency encoded in linguistic representations.*

4 Vocal-Tract Realism in Modern VC Models

4.1 Model Selection and Detection Performance

We select four VC models with different architectures and model sizes: PPG-VC Liu et al. (2021), FreeVC Li et al. (2023a), NeuralVC Cao et al. (2024), and HierSpeech++ Lee et al. (2025). Their configurations are detailed in App. B. Following the procedure in § 3, we construct a model-specific detector for each VC model and evaluate it using the same metrics as in the TTS experiments. For each model, we generate 12,000 synthetic utterances using 300 speakers from TIMIT as target speakers. We conduct five detection sessions per model, each using a different set of 50 speakers for feature selection and the remaining speakers for evaluation. Tab. 3 reports the average detection performance, together with the average number of bigram types selected per session and the total number of unique types selected across all five sessions. Fig. 2 shows the distribution of bigram selection frequencies across sessions.

Detection performance varies substantially across VC models. The detector achieves over 98% accuracy on FreeVC and NeuralVC, but less than 65% on PPG-VC and approximately 45% on HierSpeech++. Feature-selection stability follows a similar trend. For FreeVC and NeuralVC, 55.88% and 53.94% of the selected bigram types, respectively, appear in at least four of the five sessions.

These proportions decrease to 24.11% for PPG-VC and 5.3% for HierSpeech++, indicating that models with lower detection accuracy also exhibit less stable feature selection. We observe a similar pattern in the temporal coverage of selected bigrams. The detectors for FreeVC and NeuralVC tend to select more frequently occurring bigram types, with the corresponding speech segments covering an average of 62.87% and 65.26% of each utterance, respectively. In contrast, the detectors for PPG-VC and HierSpeech++ favor less frequent bigram types, yielding substantially lower coverage of 42.99% and 18.20%, respectively. These findings are consistent with our TTS evaluation: lower detection accuracy is associated with less stable feature selection and reduced temporal coverage.

Notably, despite its smaller model size, PPG-VC exhibits higher vocal-tract realism than FreeVC and NeuralVC, as suggested by its lower detection accuracy. A key architectural difference is that PPG-VC and HierSpeech++ explicitly model the fundamental frequency (F0) of the source speech and use it to guide synthesis, whereas FreeVC and NeuralVC do not. Given the potential influence of F0 on vocal-tract estimation, we hypothesize that explicit F0 modeling contributes to the increased vocal-tract realism of synthetic speech. We investigate this hypothesis through the following ablation study.

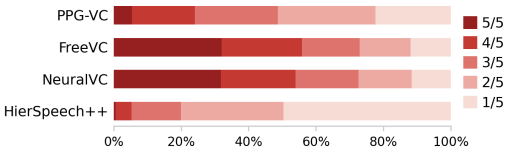

Figure 2: Bigram selection statistics of VC models.

Table 3: Detection performance on VC models.

Model	Acc.	Prec.	Rec.	FPR	Bigram (Avg. / Total)#
PPG-VC	63.42%	63.73%	61.83%	36.06%	316 / 619
FreeVC	98.82%	98.85%	98.79%	1.15%	338 / 485
NeuralVC	98.78%	98.84%	98.66%	1.10%	332 / 482
HierSpeech++	45.12%	45.06%	48.15%	57.89%	173 / 491

4.2 Ablation Study on VC Vocal-Tract Realism

VC models with explicit F0 modeling typically extract the F0 contour from the source utterance and use it to guide speech generation. Specifically, the source waveform is divided into frames, and the fundamental frequency of each frame is estimated to form an F0 sequence. This sequence is then encoded and combined with the temporally aligned content representation and target-speaker embedding to guide speech generation at each frame.

To investigate the influence of F0 on vocal-tract characteristics, we perturb the F0 sequence during speech generation and examine the resulting changes. We focus on PPG-VC and HierSpeech++, the two evaluated models that explicitly incorporate F0 into their generation pipelines, and restrict our analysis to vowel segments to obtain more reliable vocal-tract estimates. We determine the perturbation magnitude using the empirical F0 distribution of each vowel category. For each model, we first generate 10,000 utterances. Following the procedure in § 3, we identify phoneme intervals using our previously trained frame-level classifier. We then group the F0 values of frames belonging to the same vowel category and calculate their standard deviation, denoted by σ_f. Using the same source utterances and target-speaker references, we generate another 10,000 utterances with perturbed F0 sequences. For each frame belonging to a vowel, we replace its original F0 value f_o with $\hat{f} = f_o + \alpha\sigma_f$, where σ_f is the standard deviation of the corresponding vowel category and α is independently sampled for each frame from $\mathcal{U}(-M, M)$. M controls the perturbation magnitude.

Despite the perturbation, the model still generates each frame conditioned on a single F0 value and therefore produce realistic vocal-tract characteristics within individual frames. We instead examine the consistency of vocal-tract estimates across frames within the same vowel segmen: if F0 influences vocal-tract characteristics, independently perturbing the F0 values should introduce greater variation in the estimated vocal-tract features across frames. To quantify this variation, we introduce the normalized L1 distance, denoted by ℓ_1^s, between the reflection-coefficient vectors of two frames. Given two vectors V^j and V^k from the same vowel segment, we calculate $\ell_1^s(V^j, V^k) = \sum_{n=0}^{N-1} \left| \frac{v_n^j - v_n^k}{\mu_n} \right|$, where v_n^j and v_n^k denote their n-th reflection coefficients, respectively. The normalization factor μ_n is the mean of the n-th reflection coefficient across frames of the corresponding vowel category, estimated from the previously generated empirical dataset. A larger ℓ_1^s indicates greater variation in vocal-tract estimates between frames.

We compare utterances generated with two perturbation magnitudes, $M = 0.1$ and $M = 0.5$, against those generated with the original F0 sequences. Within each utterance, we pair frames belonging to the same vowel segment and calculate their ℓ_1^s distances. Compared with the unperturbed setting, $M = 0.1$ and $M = 0.5$ increase the average ℓ_1^s by factors of $1.34\times$ and $1.85\times$ for PPG-VC,

and $1.52\times$ and $2.02\times$ for HierSpeech++, respectively. Tab. 4 additionally reports the results for three randomly selected vowel categories, which exhibit similar trends. These results show that F0 perturbation increases inter-frame variation in vocal-tract estimates, with larger perturbations producing greater variation. This observation supports our hypothesis that explicit F0 modeling contributes to the vocal-tract realism of generated speech. HierSpeech++ exhibits somewhat greater sensitivity to F0 perturbation than PPG-VC, potentially reflecting differences in how the two models incorporate F0 information. However, the observed differences alone are insufficient to establish that HierSpeech++ relies more heavily on F0.

Table 4: ℓ_1^s increases on different vowels.

IPA	Model	$M = 0.1$	$M = 0.5$
/ɔ/	PPG-VC	1.46	2.1
	HierSpeech++	1.99	2.55
/u/	PPG-VC	1.21	1.47
	HierSpeech++	1.31	1.55
/aʊ/	PPG-VC	1.46	1.85
	HierSpeech++	1.47	1.97

Takeaway. *VC models conditioned on F0 sequences can generate utterances with high vocal-tract realism, making them difficult to detect. Our experimental results suggest that explicit F0 modeling plays an important role in achieving this realism. Future studies on VC deepfake should pay extra attention to how F0 information is represented and incorporated into speech generation.*

5 Post-Generation Detection Evasion

We further investigate the vulnerability of vocal-tract detection to post-generation audio editing and introduce *VoicePatch*, a novel model-agnostic evasion attack.

5.1 Adversary Model

We consider a target system, such as an online voice-chat platform, equipped with a vocal-tract-based deepfake detector. We use the bigram-based detector proposed in Blue et al. (2022) as a representative target. The attacker knows the detector's design but cannot access its deployed instance and aims to generate synthetic speech that bypasses detection. Consistent with prior studies, we assume that the attacker has access to a large real-speech dataset and a few reference utterances from the target speaker. To demonstrate the effectiveness of VoicePatch, we conservatively assume that the attacker uses a deepfake model whose unmodified outputs can be reliably detected.

5.2 VoicePatch Design

The attacker first generates synthetic speech using the deepfake model and then replaces detector-sensitive portions with acoustically similar segments from real recordings. The attack consists of two stages: offline setup and online editing. During offline setup, the attacker uses the available deepfake model to construct a synthetic-speech dataset and build a surrogate detector following the known design of the target detector. This allows the attacker to identify the bigram types likely used by the target detector. We acknowledge that the selected bigram types may differ between the surrogate and target detectors and experimentally evaluate the impact of this mismatch in § 5.3. During online editing, VoicePatch identifies occurrences of the selected bigrams in a synthetic utterance and merges overlapping segments into non-overlapping intervals, referred to as *detection units*. For each detection unit, VoicePatch retrieves a suitable replacement segment from real speech and integrates it into the synthetic utterance.

Real-speech segment selection. For each detection unit, VoicePatch searches the real-speech dataset for segments containing the same phoneme sequence. We retain candidates whose durations differ from that of the detection unit by less than 30% and rank them according to their timbre and prosody similarity. Specifically, we use pretrained FACodec Ju et al. (2024) to extract timbre and prosody representations. FACodec represents the timbre of an utterance using a 256-dimensional embedding. We calculate the timbre similarity S_t between each candidate and the detection unit as the cosine similarity of their timbre embeddings. For prosody, FACodec produces an eight-dimensional embedding every 12.5 ms. To accommodate differences in duration, we uniformly sample 16 embeddings from each sequence and calculate the average cosine similarity between corresponding embeddings, denoted by S_p. We combine the two measures into an overall similarity score, $S = 0.5S_t + 0.5S_p$, and select the candidate with the highest score. Finally, we scale the re-

placement waveform to match the average amplitude of the corresponding detection unit, reducing loudness differences.

Replacement Segment Integration. To smoothly integrate the selected real-speech segments into the synthetic utterance, VoicePatch retains additional context immediately preceding and following each replacement segment in its original recording. These context segments are blended with the synthetic speech surrounding the corresponding detection unit using dynamically varying weights. This preserves the original replacement waveform while smoothing the transitions at both boundaries. Specifically, VoicePatch applies a raised-cosine crossfade over a default 20 ms transition interval at each boundary. For the transition from synthetic to real speech, the output waveform is given by $y(t) = \frac{1+\cos(\pi t/T)}{2} x_{\mathrm{syn}}(t) + \frac{1-\cos(\pi t/T)}{2} x_{\mathrm{real}}(t)$ for $0 \leq t \leq T$, where x_{syn} denotes the synthetic speech preceding the detection unit, x_{real} denotes the additional real-speech context preceding the replacement segment, and T is the transition duration. The weights gradually shift from synthetic to real speech, producing a smooth transition into the replacement segment. The reverse weighting is applied at the other boundary to blend the additional real-speech context following the replacement segment with the synthetic speech following the detection unit. Since the detector classifies an utterance as real when at least half of the evaluated features pass their respective threshold tests, replacing every detection unit would introduce unnecessary computational overhead and audio distortion. VoicePatch therefore replaces only a subset of detection units to balance evasion performance and speech quality. Empirical evaluation identifies 60% coverage of detector-sensitive bigram segments as a favorable balance.

5.3 Evaluation of the VoicePatch Attack

We evaluate VoicePatch on RT Voice Cloning, FreeVC, and NeuralVC, all of which are reliably detected by the original vocal-tract detector. Using TIMIT and VCTK Yamagishi et al. (2019) as real-speech sources, we generate 5,000 VoicePatch-edited utterances for each model, replacing 60% of the bigram segments identified by the surrogate detector. We then evaluate the edited utterances using the corresponding detectors constructed previously. VoicePatch increases the detection bypass rates from 1.41%, 1.12%, and 1.00% to 91.11%, 93.20%, and 93.40% for RT Voice Cloning, FreeVC, and NeuralVC, respectively. These results demonstrate that VoicePatch effectively bypasses vocal-tract detection across different speech synthesis architectures.

Impact of Feature Selection Error. Although bigram selection is relatively stable for deepfake models with low vocal-tract realism, differences in the speaker sets used for feature selection may lead the attacker's surrogate detector to identify different bigrams from those used by the deployed detector. To evaluate the impact of this mismatch, we construct six detectors for each deepfake model using different speaker sets. We designate the first as the target detector and the remaining five as attacker-reproduced detectors. For each surrogate detector, VoicePatch edits synthetic utterances based on its selected bigrams, and we evaluate the resulting utterances against the target detector. The detection results presented in Tab. 5 shows that feature-selection mismatch reduces the evasion performance of VoicePatch but does not eliminate its effectiveness. Across the five surrogate detectors, VoicePatch achieves average detection bypass rates of 85.50%, 81.60%, and 85.80% for RT Voice Cloning, FreeVC, and NeuralVC, respectively. Even in the worst case, the corresponding bypass rates remain at 82.63%, 78.20%, and 83.60%. These results demonstrate that VoicePatch remains effective even when the attacker cannot precisely reproduce the bigrams selected by the deployed detector.

Speech Quality. To assess whether VoicePatch compromises speech quality, we evaluate the edited utterances in terms of naturalness, speaker similarity, and speech recognition accuracy. We measure naturalness using UTMOS Saeki et al. (2022), a non-intrusive speech-quality model that predicts mean opinion scores. For speaker similarity, we compute the cosine similarity between the speaker embeddings of each synthetic utterance and a real reference utterance from the target speaker. We extract these embeddings using the speaker encoder of the corresponding deepfake model to avoid potential inconsistencies introduced by external encoders. To evaluate speech recognition accuracy, we transcribe each utterance using Whisper Radford et al. (2023) and calculate its character error rate (CER) and word error rate (WER) against the reference text. Tab. 6 compares the quality original synthetic utterances (O) and VoicePatch-edited utterances (P) using 5,000 utterances per model. VoicePatch causes negligible degradation in MOS and speaker similarity, with minor impact on CER and WER.

Table 5: Impact of bigram selection errors.

Session	RT Voice Cloning	FreeVC	NeuralVC
1	87.07%	78.20%	86.60%
2	85.25%	79.40%	87.00%
3	85.66%	86.60%	83.60%
4	82.63%	82.20%	84.40%
5	86.87%	81.60%	87.40%
Average	**85.50%**	**81.60%**	**85.80%**

Table 6: Speech quality assessment results.

Model	MOS	Speaker sim.	CER	WER
RTVC (O)	2.48	0.78	5.45%	9.91%
RTVC (P)	2.63	0.81	6.46%	12.50%
FreeVC (O)	4.06	0.76	1.36%	3.64%
FreeVC (P)	3.67	0.78	1.64%	4.08%
NeuralVC (O)	4.05	0.78	1.91%	4.69%
NeuralVC (P)	3.66	0.78	2.17%	4.83%

Computation Overhead. We evaluate the computational overhead of VoicePatch on a server equipped with an NVIDIA H200 NVL GPU and an AMD EPYC 9455 CPU. During the offline stage, VoicePatch extracts the timbre and prosody features of potential replacement segments from the real-speech dataset. This one-time process takes approximately $68\,\mathrm{min}$, and the extracted features are reused during online editing. In the online stage, VoicePatch edits each synthetic utterance in an average of around $30\,\mathrm{ms}$, with a maximum of around $100\,\mathrm{ms}$. These results demonstrate the low online computational overhead of VoicePatch and its potential for near-real-time audio editing.

6 Related Work

Several studies have conducted literature survey or systematically evaluation on the state of learning-based audio deepfake detection Müller et al. (2022); Khanjani et al. (2023); Li et al. (2025b); Firc et al. (2025). For instance, Müller et al. constructed an in-the-wild dataset and evaluated representative learning-based detection architectures Müller et al. (2022). Yi et al. presented a comprehensive survey of audio deepfake generation and detection, and conducted unified evaluations of representative features and classifiers Yi et al. (2023). More recently, Li et al. developed the SONAR benchmark and evaluated many existing learning-based detector Li et al. (2025b). However, these studies primarily consider learning-based or conventional signal-processing detectors and do not systematically evaluate vocal-tract detection against modern TTS and VC systems.

Extensive studies have also investigated attacks against audio deepfake detection. For example, optimization-based audio perturbations have been explored in Kawa et al. (2023); Farooq et al. (2025). Learning-based suppression of synthesis artifacts has been studied in Panariello et al. (2023). Lightweight manipulations, including noise injection, volume adjustment, and fading, have also been evaluated as practical evasion strategies Wu et al. (2026). However, these attacks primarily target neural detectors rather than vocal-tract detection. Not guided by the vocal-tract detection principle, those audio manipulations may introduce additional vocal-tract inconsistencies. The original vocal-tract detection study considered two adaptive attack strategies but found them either ineffective or computationally impractical Blue et al. (2022).

7 Conclusion

This paper revisited the reliability of vocal-tract-based audio deepfake detection against modern TTS and VC systems. Our evaluation shows that vocal-tract detection performs substantially worse on recent generation models than on earlier synthesis approaches, often yielding low accuracy and impractically high false-positive rates. Through extensive ablation studies, we further identify learned linguistic representations in TTS models and explicit source-prosody modeling in VC models as major contributors to the realistic vocal-tract estimates observed in synthetic speech. We also introduce a post-generation editing attack that replaces detection-relevant synthetic segments with acoustically matched real-speech clips and smooths the resulting transitions to preserve acoustic naturalness. The attack is applicable to different generation pipelines and can dramatically increase the likelihood that synthetic speech is classified as genuine. Overall, these findings expose fundamental limitations of vocal-tract detection and underscore the need for more robust audio deepfake defenses that remain effective against both advanced generation models and post-generation manipulation.

AI USE STATEMENT

The authors used AI tools solely to polish the writing of this manuscript. AI tools were not used for coding, technical design, or development of the conceptual framework. The authors reviewed and take responsibility for the final content.

ETHICS STATEMENT

VoicePatch replaces segments of synthetic speech with real speech to evade a detector that relies on vocal-tract-related physiological cues. This method could be misused to conceal manipulated speech or enable deceptive impersonation. Our experiments used existing TIMIT and VCTK recordings; we collected no new speech recordings and recruited no participants for listening tests or annotation. We plan to release code for segment selection and local boundary smoothing, but will not redistribute original or processed recordings. This release may help researchers evaluate the limitations of speech deepfake detectors, while also making the method available for misuse. We oppose deceptive use and unauthorized impersonation, and urge users to respect speakers' privacy and the permissions governing the recordings they use.

REPRODUCIBILITY STATEMENT

The real-speech segment selection and local boundary smoothing procedures are described in Section 5. The detector is based on the publicly available implementation of Blue et al. (2022). An anonymized implementation of the detector and proposed method is included with this submission.

REFERENCES

Moustafa Alzantot, Ziqi Wang, and Mani B Srivastava. Deep residual neural networks for audio spoofing detection. In *Proc. Interspeech*, pp. 1078–1082, 2019. doi: 10.21437/Interspeech.2019-3174.

Hussam Azzuni and Abdulmotaleb El Saddik. Voice cloning: Comprehensive survey. *arXiv:2505.00579*, 2025.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: a framework for self-supervised learning of speech representations. In *International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 12449–12460, 2020.

Logan Blue, Kevin Warren, Hadi Abdullah, Cassidy Gibson, Luis Vargas, Jessica O'Dell, Kevin Butler, and Patrick Traynor. Who are you (I really wanna know)? detecting audio DeepFakes through vocal tract reconstruction. In *USENIX Security Symposium*, Aug. 2022.

Danyang Cao, Zeyi Zhang, and Jinyuan Zhang. Neuralvc: Any-to-any voice conversion using neural networks decoder for real-time voice conversion. *IEEE Signal Processing Letters*, 31:2070–2074, 2024.

Edresson Casanova, Julian Weber, Christopher Dane Shulby, Arnaldo Candido Junior, Eren Gölge, and Moacir Antonelli Ponti. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone. In *International Conference on Machine Learning (ICML)*, volume 162, pp. 2709–2720, 2022.

Edresson Casanova, Kelly Davis, Eren Gölge, Görkem Göknar, Iulian Gulea, Logan Hart, Aya Aljafari, Joshua Meyer, Reuben Morais, Samuel Olayemi, et al. Xtts: a massively multilingual zero-shot text-to-speech model. *arXiv:2406.04904*, 2024.

Muhammad Umar Farooq, Awais Khan, Kutub Uddin, and Khalid Mahmood Malik. Transferable adversarial attacks on audio deepfake detection. In *Winter Conference on Applications of Computer Vision (WACV) Workshops*, pp. 1640–1649, Feb. 2025.

Anton Firc, Kamil Malinka, and Petr Hanáček. Evaluation framework for deepfake speech detection: a comparative study of state-of-the-art deepfake speech detectors. *Cybersecurity*, 8, 2025.

John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. Darpa timit acoustic-phonetic continuous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon technical report*, 93:27403, 1993.

Corentin Jemine. Real-Time Voice Cloning. `https://github.com/CorentinJ/Real-Time-Voice-Cloning`, 2019. GitHub repository.

Ye Jia, Yu Zhang, Ron J Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Advances in neural information processing systems*, volume 31, pp. 4480–4490, 2018.

Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Eric Liu, Yichong Leng, Kaitao Song, Siliang Tang, Zhizheng Wu, Tao Qin, Xiangyang Li, Wei Ye, Shikun Zhang, Jiang Bian, Lei He, Jinyu Li, and Sheng Zhao. NaturalSpeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 22605–22623. PMLR, 2024.

Piotr Kawa, Marcin Plata, and Piotr Syga. Defense against adversarial attacks on audio deepfake detection. In *Proc. Interspeech*, pp. 5276–5280, 2023.

Zahra Khanjani, Gabrielle Watson, and Vandana P Janeja. Audio deepfakes: A survey. *Frontiers in Big Data*, 5, 2023. doi: 10.3389/fdata.2022.1001063.

Jaehyeon Kim, Jungil Kong, and Juhee Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International conference on machine learning*, pp. 5530–5540. PMLR, 2021.

Sang-Hoon Lee, Ha-Yeong Choi, Seung-Bin Kim, and Seong-Whan Lee. Hierspeech++: Bridging the gap between semantic and acoustic representation of speech by hierarchical variational inference for zero-shot speech synthesis. *IEEE Transactions on Neural Networks and Learning Systems*, 36(10):18422–18436, 2025.

Jingyi Li, Weiping Tu, and Li Xiao. Freevc: Towards high-quality text-free one-shot voice conversion. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023a.

Menglu Li, Yasaman Ahmadiadli, and Xiaoping Zhang. A survey on speech deepfake detection. *ACM Computing Surveys*, 57(7):1–38, 2025a. doi: 10.1145/3714458.

Xiang Li, Pin-Yu Chen, and Wenqi Wei. Where are we in audio deepfake detection? a systematic analysis over generative and detection models. *ACM Transactions on Internet Technology*, 25(3): 20:1–20:19, 2025b.

Yinghao Aaron Li, Cong Han, Vinay Raghavan, Gavin Mischler, and Nima Mesgarani. Styletts 2: Towards human-level text-to-speech through style diffusion and adversarial training with large speech language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 19594–19621, 2023b.

Songxiang Liu, Yuewen Cao, Disong Wang, Xixin Wu, Xunying Liu, and Helen Meng. Any-to-many voice conversion with location-relative sequence-to-sequence modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1717–1728, 2021.

Zohreh Mostaani and Mathew Magimai Doss. On breathing pattern information in synthetic speech. In *Proc. Interspeech*, pp. 2768–2772, 2022.

Nicolas Müller, Pavel Czempin, Franziska Dieckmann, Adam Froghyar, and Konstantin Böttinger. Does audio deepfake detection generalize? In *Proc. Interspeech*, pp. 2783–2787, 2022.

Zhikang Niu, Shujie Hu, Jeongsoo Choi, Yushen Chen, Peining Chen, Pengcheng Zhu, Yunting Yang, Bowen Zhang, Jian Zhao, Chunhui Wang, et al. Semantic-vae: Semantic-alignment latent representation for better speech synthesis. *arXiv:2509.22167*, 2025.

Michele Panariello, Wanying Ge, Hemlata Tak, Massimiliano Todisco, and Nicholas Evans. Malafide: a novel adversarial convolutive noise attack against deepfake and spoofing detection systems. In *Proc. Interspeech*, pp. 2868–2872, 2023.

Lawrence R. Rabiner and Ronald W. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall, 1978.

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 (ICML)*, volume 202, pp. 28492–28518, 2023.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. Utmos: Utokyo-sarulab system for voicemos challenge 2022. In *Proc. Interspeech*, pp. 4521–4525, 2022.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerry-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 4779–4783, 2018.

Kai Shen, Zeqian Ju, Xu Tan, Eric Liu, Yichong Leng, Lei He, Tao Qin, Jiang Bian, et al. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. In *International Conference on Learning Representations (ICLM)*, Jul. 2024.

Berrak Sisman, Junichi Yamagishi, Simon King, and Haizhou Li. An overview of voice conversion and its challenges: From statistical modeling to deep learning. *IEEE/ACM transactions on audio, speech, and language processing*, 29:132–157, 2020.

Catherine Stupp. Fraudsters used AI to mimic CEO's voice in unusual cybercrime case. `https://perma.cc/9RF5-G72R`.

Nishant Subramani and Delip Rao. Learning efficient representations for fake speech detection. In *AAAI Conference on Artificial Intelligence*, volume 34, pp. 5859–5866, 2020.

Xiaohai Tian, Rohan Kumar Das, and Haizhou Li. Black-box attacks on automatic speaker verification using feedback-controlled voice conversion. In *Proc. Odyssey*, pp. 159–164, 2020.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv:2301.02111*, 2023.

Haolin Wu, Jing Chen, Ruiying Du, Cong Wu, Kun He, Xingcan Shang, Hao Ren, and Guowen Xu. Clad: Robust audio deepfake detection against manipulation attacks with contrastive learning. *Knowledge-Based Systems*, 335, 2026.

Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald. CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92), 2019.

Jiangyan Yi, Chenglong Wang, Jianhua Tao, Xiaohui Zhang, Chu Yuan Zhang, and Yan Zhao. Audio deepfake detection: A survey. *arXiv:2308.14970*, 2023.

A Vocal-Tract Detector Implementation

We implement the detector according to Blue et al. (2022). The detector approximates the vocal tract as $N = 2LF/C$ equal-length, lossless tube sections, where L, F, and C denote the vocal-tract length, audio sampling rate, and speed of sound, respectively. We set L to the average human vocal-tract length of $15.5\,\mathrm{cm}$. The detector estimates tube cross-sectional areas from speech segments containing two adjacent phonemes, referred to as bigrams. Following the original design, each bigram is divided into $35\,\mathrm{ms}$ frames with a $7\,\mathrm{ms}$ overlap between adjacent frames. For each frame, the detector iteratively adjusts the reflection coefficients $(r_0, \ldots, r_{N-1})$ until the resulting spectrum sufficiently matches the observed spectrum. Using the average human glottal cross-sectional area of

$3.7\,\mathrm{cm}^2$ as a reference, the detector converts the estimated reflection coefficients into tube-area estimates. It then aggregates the estimated areas of individual tube sections across multiple occurrences of each bigram type to characterize their normal ranges in real speech.

Following Blue et al. (2022), we select features for detection a real-speech dataset and synthetic utterances generated by the target deepfake model. For each bigram type, we extract its occurrences from both datasets and estimate their tube-area profiles. A tube-section area associated with a particular bigram type is selected as a detection feature if a threshold can distinguish synthetic from real speech with both precision and recall of at least 0.9. Each selected feature is therefore defined by a bigram type, a tube-section index, and a corresponding decision threshold. To classify an utterance, the detector identifies occurrences of the selected bigram types, estimates their tube-area profiles, and applies the corresponding feature thresholds. The utterance is classified as real if at least half of the evaluated features pass their respective threshold tests; otherwise, it is classified as synthetic.

To validate our implementation, we reproduce the evaluation setting of Blue et al. (2022), using TIMIT Garofolo et al. (1993) as the real-speech dataset and Real-Time Voice Cloning Jemine (2019) to generate synthetic speech. We select 300 speakers from TIMIT and generate approximately 30,000 corresponding synthetic utterances, each containing a complete sentence. In each experimental session, we use the real and synthetic utterances associated with 50 speakers for feature selection and those associated with the remaining 250 speakers for evaluation. We repeat the experiment five times, selecting a different set of 50 speakers in each session. On average, the detector achieves an accuracy of 99.48%, a precision of 99.9%, and a recall of 99.5%. These metrics are calculated as $\mathrm{Accuracy} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$, $\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$, and $\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$, respectively, where TP, TN, FP, and FN denote the numbers of true-positive, true-negative, false-positive, and false-negative samples. We treat synthetic speech as the positive class. The resulting performance is close to that reported in Blue et al. (2022), confirming that the detector is properly implemented.

B Configuration of The Evaluated Deepfake Models

Table 7: Configurations of the selected TTS models.

Model	Architecture	Size	Emb. Dim.
YourTTS (2022)	Transformer & GAN	38M	192
StyleTTS2 (2023)	Diffusion & GAN	136M	512
XTTS (2024)	Transformer & VAE & GAN	467M	1024

Table 8: Configurations of the selected VC models.

Model	Architecture	Size	Emb. Dim.
PPG-VC (2021)	Seq2seq & attention	36M	144
FreeVC (2023)	VAE & GAN	347M	192
NeuralVC (2024)	VAE & GAN	123M	192
HierSpeech++ (2025)	VAE & flow & GAN	381M	192

C Impact of Replacement Proportion

This experiment evaluates the detection bypass rate and speech quality at different proportions of replaced detector-sensitive bigram segments to identify a favorable trade-off between evasion effectiveness and speech quality. The results are presented in Tab. 9 .

Table 9: Detection bypass rate under different replacement Proportions.

Model / Coverage	Pass Rate	UTMOS	Speaker Sim.	CER	WER
RTVC / 0% (unedited)	1.41%	2.48	0.82	5.45%	9.91%
RTVC / 55%	87.47%	2.62	0.81	6.19%	12.15%
RTVC / 60%	91.11%	2.63	0.81	6.46%	12.50%
RTVC / 65%	91.82%	2.67	0.81	6.77%	12.88%
RTVC / 70%	93.33%	2.71	0.82	6.98%	13.28%
FreeVC / 0% (unedited)	1.12%	4.06	0.76	1.36%	3.64%
FreeVC / 55%	89.80%	3.69	0.78	1.60%	3.99%
FreeVC / 60%	93.20%	3.67	0.78	1.64%	4.08%
FreeVC / 65%	95.00%	3.66	0.78	1.73%	4.18%
FreeVC / 70%	95.80%	3.64	0.78	1.62%	3.94%
NeuralVC / 0% (unedited)	1.00%	4.05	0.78	1.91%	4.69%
NeuralVC / 55%	89.00%	3.68	0.78	2.09%	4.79%
NeuralVC / 60%	93.40%	3.66	0.78	2.17%	4.83%
NeuralVC / 65%	95.20%	3.65	0.78	2.10%	4.81%
NeuralVC / 70%	96.80%	3.63	0.78	2.21%	4.69%