

An LLM-enabled End-to-End Attack Pipeline against Speech-based Machine Translation

Junyi Zhao¹, Zeyan Liu², Zijun Yao¹, Fengjun Li¹, and Bo Luo¹

¹ University of Kansas, Lawrence, KS 66045, USA

² University of Louisville, Louisville, KY 40208, USA

{junyi.zhao, zyao, fli, blau}@ku.edu; zeyan.liu@louisville.edu

Abstract. With the growing demand for global communication, speech-based Machine Translation (MT) has become an indispensable tool for bridging language barriers. Despite their widespread use, the security of speech-based MT systems has not been adequately investigated, especially in the context of adversarial AI, where attackers deliberately manipulate the input speech to disrupt the system’s outputs. In this paper, we systematically examine the security of speech-based MT systems. We identify key attack strategies, assess their effectiveness under various scenarios, and highlight the challenges in mitigating these threats. Through extensive experiments and a user study, we demonstrate the real-world feasibility and impact of these attacks. Our findings emphasize the critical need for robust defense mechanisms and call for increased research focus on enhancing the security of machine translation technologies.

1 Introduction

Machine translation (MT) has evolved significantly since its inception in the 1950s. Early rule-based systems relied on manually crafted linguistic rules and bilingual dictionaries for word-for-word translations [32, 29, 31]. Statistical machine translation (SMT) uses large parallel corpora and probabilistic models to improve fluency and accuracy [19, 17]. Neural machine translation (NMT) uses DNNs to model complex linguistic patterns, and transformer-based architectures dominate the field to deliver high-quality, real-time translation [34, 12, 42, 21].

Speech-based Machine Translation (MT) systems automatically convert the speech in one language into speech or text in another, to facilitate seamless cross-lingual communication. With the widespread use of voice/video messages on online platforms, speech-based MT technologies became indispensable tools for global social networking. NMT models in modern speech-based MT systems can capture complex linguistic and semantic nuances through large-scale pre-training [39], which allows a robust generalization across diverse languages and domains [5]. By integrating speech recognition into NMT, these systems enable a natural way of interaction, where users speak in their native language and instantly receive translations in another through mobile devices.

However, the rapid adoption of speech-based MT brings security concerns, particularly their vulnerability to adversarial attacks. Adversaries may inject

carefully crafted perturbations (“noise”), which are often imperceptible to human listeners, into audio inputs to mislead the speech recognition module into generating incorrect or even harmful content. Such attacks can introduce a severe impact on system reliability and user trust. Despite the urgency of understanding these vulnerabilities, research on adversarial security in speech-based MT has lagged behind the development of translation models themselves. In this paper, we examine the adversarial robustness of speech-based neural machine translation systems. Specifically, we address the following research questions:

- **RQ1:** What attack vectors can be exploited in speech-based MT systems?
- **RQ2:** How can external adversaries manipulate the systems to produce (targeted) incorrect and malicious translations?
- **RQ3:** Can users defend against such attacks by manual examinations?

To answer these questions, we propose an end-to-end adversarial attack pipeline. We first convert speech to text using Automatic Speech Recognition (ASR), then identify the most critical words in the transcript. Leveraging large language models (LLM, e.g., ChatGPT), we craft prompts to adversarially replace one target word per attack. Finally, we synthesize adversarial audio by introducing subtle perturbations corresponding to the chosen replacement.

Our **main contributions** are as follows: (1) We present the first investigation of the security of speech-based machine translation pipeline, and introduce an effective adversarial attack; (2) we systematically examine and evaluate various attack strategies; and (3) we conduct a user study, which demonstrates the stealthiness and effectiveness of our adversarial manipulations.

Ethics Statement. All experiments were carried out in isolated and controlled environments to ensure safety and reproducibility. No real-world systems, users, or services were attacked or negatively impacted. The user study was approved by the Human Research Protection Program at the University of Kansas.

2 System Model and Threat Model

The System Model. We assume a speech-based general-purpose MT system, as shown in Figure 1 (A). It takes voice input (S) from the physical world and invokes its embedded speech recognition module to obtain the source text (D_s) in the source language. The system then utilizes its NMT module ($T()$) to convert D_s into destination language text (D_t), i.e., $D_t = T(D_s)$, which could be converted to voice and played to the recipient of the communication.

The Threat Model. As shown in Figure 1 (B), the attacker alters S into S' , which produces D'_s (by the benign voice recognition system). D'_s is then translated (by the benign MT system) into D'_t . The attacker’s ultimate goal is to jeopardize the communication by generating a D'_t that is semantically different from D_t , i.e., to maximize the semantic differences between D_t and D'_t .

We assume that an attacker has the following capabilities and in-capabilities: (1) We assume that the attacker has no knowledge beyond the speech recognition model, i.e., the text-to-text translation module is considered a black box. This matches the industrial realities, as most commercial translation models

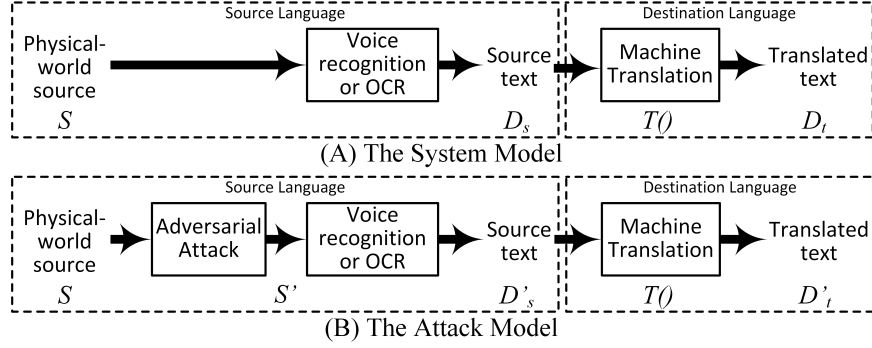


Fig. 1. The system model and the attack model.

are closed-source. Also, such translation systems often collect input from susceptible handheld/IoT devices, e.g., smartphones, while speech recognition and translation are performed on the cloud, which is better secured and easier for the large-scale language model to load. (2) The attacker has full knowledge of the speech recognition model, as such models are usually open-source. (3) The attacker cannot alter any parameters of any model, including both speech recognition and text translation. (4) The attacker can analyze and alter the input audio. However, the alteration must be made with minimum acoustic impact on the original audio to minimize unnecessary alert from the user.

We assume the victims are genuine MT users: (1) They attempt to communicate in a language they do not understand, so that they are unable to identify the maliciously altered text/speech in the destination language. (2) The recipient of the translated language may be alerted by extremely erroneous output, e.g., when the translation significantly differs from the *context* of the communication. That is, both the source- and target-language listeners rely on the MT system due to limited or no proficiency in the other language. Consequently, they have no linguistic basis to evaluate the semantic fidelity of the translation, and cannot distinguish between a translation artifact, natural ambiguity, or an adversarially induced substitution. This assumption is consistent with the primary real-world use cases of MT systems, such as cross-lingual communication between non-bilingual parties and real-time subtitle generation, where no feedback channel exists between the listener and the source-language speaker. Finally, we assume that the MT system provides reasonably high accuracy, which is achievable with state-of-the-art ASR and deep-learning (DL) based MT systems, and that the original audio is properly recognized by the ASR system.

3 An LLM-enabled Attack against Speech-based NMT

3.1 Attack Overview and Challenges

We propose an end-to-end attack framework targeting speech-based NMT systems. As illustrated in Figure 2, the framework consists of three main steps: (1) *Target term identification*: We first identify the term that contributes most

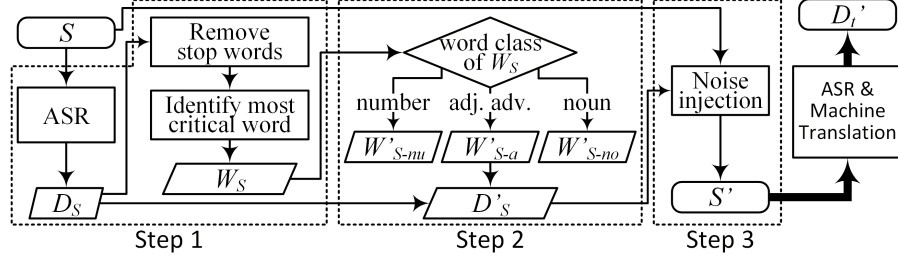


Fig. 2. Overview of the proposed attack. Step 1: Target word identification, Step 2: Replacement word search, Step 3: Adversarial noise injection.

significantly to the sentence’s overall semantics. Targeting this key term is most likely to induce a substantial change to the sentence’s meaning. (2) *Substitution term selection*: We search for a malicious replacement for the target term, with two selection objectives: (i) *effectiveness*, to significantly alter the sentence’s intended meaning, and (ii) *stealthiness*, to retain phonetic similarity to the original term to evade human detection. (3) *Adversarial noise injection*: Finally, we inject carefully crafted adversarial noise into the original audio S to produce the adversarial example S' . This perturbation ensures the substitution is realized at the audio level while remaining inconspicuous.

For an attack to be stealthy, the manipulated audio must sound natural to human listeners by avoiding obvious artifacts or distortions that may arouse suspicion. Thus, the *first* challenge is to craft adversarial inputs that are acoustically inconspicuous and semantically deceptive after passing through the speech recognition and MT. The *second* challenge is computational efficiency. Transforming an entire sentence adversarially requires significantly more computation than altering a few words, due to the larger search space that grows exponentially with the sentence length. Hence, we propose to identify the most semantically significant term(s) in the sentence and substitute such term(s) only.

Traditional attacks on natural language processing (NLP) models often follow a multi-step pipeline to identify word substitutions: (1) querying tokenization/encoding dictionaries to enumerate candidates; (2) consulting phonetic resources (e.g., CMUdict) to enforce pronunciation similarity; and (3) filtering by contextual semantics to retain plausible substitutions. The process is computationally expensive, especially when the pool of candidates grows. As a result, the identified replacement word may not be optimal. Instead, we employ an LLM as a heuristic engine to identify candidates that meet multiple constraints: the same PoS, semantic inversion, and phonetic proximity. This approach significantly reduces the search space and eliminates the need for dictionary lookup and rule-based phonetic comparisons, thereby enabling a more scalable attack.

3.2 Target Word Identification

When a text sentence D_s is extracted from the audio, we first identify the most critical word within the sentence to maximize the effectiveness of the attack. We employ two different measurements, BERT and ChatGPT’s embedding module

text-embedding-ada-002, to estimate the semantic and vector-level significance of each word in the sentence. Our method involves the following steps:

- **Stop Word Filtering and POS Selection:** Common stop words are filtered out, as they do not contribute meaningfully to the attack. We further narrow down the candidate words to specific POS: nouns (names and numbers), verbs, adjectives, and adverbs, as they tend to carry more meaningful content. We employ the Natural Language Toolkit for stop word list and POS tagging.
- **Word Removal for Impact Analysis:** For each of the remaining words w_i in the sentence, we temporarily remove it to obtain a modified sentence $D_s^{-w_i}$.
- **Similarity Measurement:** We employ two models to compute the embedding space similarity of the original sentence D_s and the modified sentence $D_s^{-w_i}$, which reflects how much semantic information is kept after removing w_i . We adopted BERT (xlm-roberta-large (English) [11]) and OpenAI’s text-embedding-ada-002. For BERT, we used the combined BERTScore measure F1 (F_{BERT} in [45]), which leverages contextual embeddings from pre-trained BERT models (RoBERTa in our case) to measure the semantic similarity between sentences.

For text-embedding-ada-002, we used the embedding space cosine similarity.

- **Target Word Selection:** We identify two candidates w_s out of all w_i : the top-1 word from BERT and text-embedding-ada-002, respectively, whose removal causes the largest drop in sentence embedding similarity between D_s and $D_s^{-w_i}$. These two words (overlapping in 19% of the cases) are considered the most impactful in D_s , and are thus selected as the attack targets. We retain both candidates for now and further discuss their impacts in Section 4.4.

Example 1. For sentence “*Could you please provide a map of the local area*”, the attacker calculates the BERT and ADA similarity scores between D_s and $D_s^{-w_i}$ for each w_i . As shown below, they both find the most important word to be “map” (BERT similarity 0.7199 and embedding-ada-002 similarity 0.9678).

w_i	please	provide	map	local	area	w_i	please	provide	map	local	area
BERT	.9669	.9830	.7199	.9198	.9266	ada-002	.9939	.9892	.9678	.9803	.9817

3.3 Searching for the Replacement Word

In the proposed attack, it is crucial to tailor the replacement word to the grammatical role and semantic category of the target word (Step 2 in Figure 2).

- We substitute adjectives and adverbs with their antonyms, to reverse or significantly alter the sentence’s meaning while preserving grammatical correctness.
- Numbers are substituted with a significantly different value that still fits the context logically (e.g., replacing “ten” with “eighty” when referring to quantity).
- Proper nouns (e.g., names of people) are replaced with other names within the same semantic domain, e.g., we replace the name of a well-known musician with a scientist. This preserves the contextual relevance while introducing variation.
- We replace common nouns and verbs with words that are semantically plausible but sufficiently different, often using synonyms, hypernyms, or related terms.

In the implementation, we prompt an LLM with instructions that specify the handling of each word category. The prompt includes a list of word types and one

representative example for each, to demonstrate the substitution strategy. This guided prompt helps the LLM generate replacements that are context-aware, semantically coherent, and effective for adversarial purposes. In this step, we find the replacement W'_{s-nu} , W'_{s-a} , or W'_{s-no} based on the category of W_s . The replacement word is generalized as W'_s . In Section 4.4, we articulate how the number/selection of the replacement words impacts attack performance.

Example 2. Continuing with Example 1, the candidate word “map” is a noun, hence, we look for replacements that are: (1) also nouns, (2) have similar pronunciation as “map”, and (3) have very different meanings. ChatGPT identified the replacement for “map” as “trap”.

3.4 Adversarial Noise Injection

To improve efficiency, we restrict the perturbation to only the target word, not modifying the entire sentence. The noise injection approach is as follows:

- **Audio Segmentation:** Based on the timing information (e.g., obtained via forced alignment or from the ASR model’s timestamp output), we accurately extract the audio segment corresponding to the selected target word w_s .
- **Adversarial Perturbation:** The extracted segment is then passed into an adversarial attack module adapted and modified from [9]. In each iteration, the adversarial perturbation in the audio clip is adjusted (guided by gradient descent), and the audio clip with modified perturbation is evaluated by the speech recognition module to check if the text output is altered as designed. The goal is to push the ASR model to misclassify the perturbed segment while keeping the perturbation imperceptible to human listeners.
- **Audio Reconstruction:** We replace the original segment of w_s in the full audio with the perturbed version by concatenating the adversarial segment back into the original stream. This localized attack ensures that only a minimal portion of the audio is modified, which not only improves computational efficiency but also reduces the risk of degrading the naturalness of the rest of the audio.
- **[Optional] Adding White Noise.** To cover up the adversarial noise added to the target word, we optionally inject low-level white noise into the rest of the audio clip, i.e., the non-target words. In practice, adding white noise in the range of -25dB to -10dB can serve as an effective masking signal, helping the adversarial perturbation blend more naturally into the overall audio.

Example 3. Continuing with Example 2, segments “area” and “map” are extracted from the audio. Adversarial perturbations are injected to trick the ASR module to classify them as “aura” and “trap”. We integrate these perturbed fragments back to the original audio to produce clips of “Could you please provide a map of the local *aura*” and “Could you please provide a *trap* of the local area”.

3.5 Prompt Engineering

The perturbation task involves modifying words across a range of lexical types, each of which requires a distinct transformation strategy: while adjectives and

adverbs often allow for straightforward semantic inversion through antonyms (e.g., happy→miserable, quickly→slowly), other categories (proper nouns, numbers) do not have natural antonymic counterparts. Instead, their perturbation might involve substitution based on categories. Given this diversity, a one-size-fits-all prompting approach is insufficient.

Few-shot prompts are known to enhance LLMs’ ability to perform complex tasks [7]. We adopt a multi-type few-shot framework to provide the LLM with distinct examples of each type. This allows the model to learn a generalized mapping between word types and their corresponding perturbation logic, even in cases where the transformation is more abstract. All transformations preserve the lexical category of the target word to ensure that the output remains grammatically coherent within the original sentence structure. This approach not only improves the model’s ability to execute diverse transformations reliably, but also encourages semantic and syntactic stability, both of which are critical for downstream tasks. By distinguishing and applying context-sensitive operations across word types, we enable more robust and linguistically grounded perturbations.

4 Experiments

4.1 Settings

We implemented the attack in Python 3.7.16. We used OpenAI’s API for GPT-4 [3] and text embedding. We randomly selected 10,000 questions from the SQuAD Dataset [28] and converted text questions into audio clips using Google gTTS [1]. Unless otherwise specified, the default source/destination languages are English and German, and default translator is Google Translate. All GPU experiments were conducted on an Ubuntu desktop with Intel Xeon W5-3423 CPU and RTX A6000 GPU, while the others used an Apple M1 Max 64G laptop.

We implemented a fully automated end-to-end attack pipeline as described in Section 3. The target word selection, replacement word search, and LLM prompt engineering components are completely new. The audio perturbation module builds upon the open-source implementation of [9] with the following improvements: (1) the iterations are optimized to improve efficiency, (2) the perturbation backend is rewritten to utilize modern GPU acceleration (their implementation only uses CPU). All other peripheral modules, including ASR integration and translation backend interface, are also implemented by us.

The Baselines. To our knowledge, no off-the-shelf baseline exists for end-to-end attacks on speech NMT. We therefore assembled relevant techniques in the literature to construct baselines: (A) For each input sentence, we choose a word with an antonym in WordNet [22] as the target and the antonym as replacement. The original attack in [9] is employed to generate the adversarial perturbation. (B) The targeted flip (TF) attack in [37] is a targeted text-to-text attack that changes input tokens to the NMT system to flip its output. (C) The universal trigger attack (UTA) in [37] is a more efficient, untargeted T2T attack that tricks the NMT to generate any arbitrary malicious output. Note that TF and UTA

Table 1. Comparison of our attack against the baseline. ASR: Attack Success Rate at 50 iterations. Embedding: The embedding similarities between the clean and adversarial translations. Translation from English to German. (*: one-time cost, not per-instance.)

Methods	ASR $\uparrow$	Efficiency	Stealth	Effectiveness					
		Time(s) $\downarrow$	NMSE $\downarrow$	chrF $\downarrow$	ada $\downarrow$	BLEU $\downarrow$	Rouge2 $\downarrow$	RougeL $\downarrow$	CIDEr $\downarrow$
Benign	-	-	0	100	1	100	1	1	10
A. Antonym	71.7%	550-750	3.8×10^6	95.1	0.978	77.40	0.827	0.888	7.86
B. TF [37]	18.1%	7.97	text-only	53.5	0.909	33.25	0.411	0.585	3.02
C. UTA [37]	100%	28*	text-only	9.1	0.178	5.96	0.102	0.188	1.01
Ours	98.9%	CPU: 60-80 GPU: 8-12	1.4×10^3	60.0	0.694	38.60	0.339	0.539	3.37

in [37] are text-domain attacks that modify tokens in the source text. They offer no stealth as the text is visibly altered and obviously malicious.

4.2 Main Results

We first evaluate the proposed attack on five key aspects: cost, attack success rate, real-world efficiency, and effectiveness of successful attacks.

Cost. At the time of initial writing (2025/09), OpenAI charges \$1.25/\$10.00 (input/output) per 1M tokens for GPT-5. In our attack, input and output would not exceed 100 tokens per attack. Therefore, each attack costs less than 0.2 cents.

Attack Success Rate (ASR). ASR is defined as the proportion of adversarial examples that successfully change the translation outputs. Our method achieves 98.9% ASR in no more than 50 iterations. That is, it generates strong attacks even with limited computational budgets. In comparison, Baseline (A) only reaches 71.7% ASR at 50 iterations and needs nearly 1,000 iterations to match our ASR. This is because their target and replacement words often have low acoustic similarity, making it harder to create adversarial audio examples. The text-based attack in baseline (B) achieves only 18.1% ASR in 7.97 seconds. Baseline (C) is an untargeted attack that is always capable of producing an arbitrary malicious translation. It takes 28 seconds (one-time cost) to generate the adversarial token string, which could be applied to any text input.

Attack Efficiency. We tested the proposed attack under ten parallel processes to simulate a realistic deployment. The primary overhead is the iterative calculation of *Connectionist Temporal Classification (CTC)* results and multiple directed attacks by gradient descent. This procedure requires thousands of back-propagations through unrolled Recurrent Neural Networks and is further dragged by the algorithm’s binary search loop for fine-tuning the regularization constant. To mitigate this, we designed/implemented three optimization mechanisms:

- As described in Section 3.4, we aim to alter only one word from the original sentence; therefore, we only inject adversarial noise into the audio segment for the target word, to save computation from full-sentence perturbation.
- In identifying the replacement word, we explicitly prompt the LLM to find an antonym with *similar pronunciation*, so that our attack required significantly fewer iterations to generate each successful adversarial example. With these two algorithmic optimizations, each attack took approximately 60 to 80 seconds.

- The attack in [9] only uses CPU for adversarial noise injection. We have completely re-written the perturbation code to make full use of the GPU.

With these optimizations, our attack generates ten adversarial examples in 80 to 120 seconds, with an average of 8-12 seconds per example. In contrast, baseline (A) needs 550-750 seconds per example (using CPU only). This results in a significant speedup without compromising attack effectiveness. This further highlights the practicality and scalability of our attack. Although Baseline (B) is a text-only attack, its efficiency is similar to our audio attack.

Attack Effectiveness. We evaluate how the attack changes the semantic meaning of the translated text using five metrics: (1) Text Embedding Similarity (Ada-002), (2) Character-level F-score (chrF), (3) Bilingual Evaluation Understudy (BLEU), (4) Recall-Oriented Understudy for Gisting Evaluation (ROUGE). We adopted ROUGE-2 and ROUGE-L, and (5) Consensus-based Image Description Evaluation (CIDEr). In our evaluations, CIDEr captures semantic adequacy and penalizes generic/trivial outputs, thereby complementing metrics such as BLEU and ROUGE with a stronger emphasis on content relevance and diversity. The measurements of two identical sentences (i.e., no attack) are shown in Table 1 Row “Benign”. Details of the metrics are provided in Appendix A.

As shown in Table 1, all of the chrF score, ada-002 embedding similarity, BLEU, ROUGE, and CIDEr are significantly lower for adversarial translations generated by our attack, compared to Baseline (A). Baseline (B) in [37] did not specify a mechanism to identify target/replacement words, hence, we applied our methods for word selection in the experiments. That is, in their successful attacks, the target and replacement words are the same as our attack, yielding similar effectiveness. Meanwhile, Baseline (C) generates arbitrary, ungrammatical target-language snippets that significantly deviate from the benign translation. Although similarity scores are low, the translations are obviously malicious and nonsensical. That is, our method introduces larger semantic distortion to increase the likelihood of user misunderstanding in more practical settings.

Additional Languages. We further evaluate the attack with three categories of languages: (1) major (resourceful) languages: French (fr), Italian (it), Spanish (es); (2) low-resource languages: Irish (ga), Afrikaans (af), Swahili (sw), Somali (so); and (3) morphologically rich language: Turkish (tr). “Morphologically rich” languages have words with extensive inflectional structure, where a single lemma yields many surface forms, which complicates tokenization, lemmatization, and modeling. “Low-resource” denotes languages with limited digital data and tools, e.g., with few balanced annotated corpora; scarce tokenizers/analyzers/parsers.

The attack performance is presented in Table 2. We did not measure the BERTScore for the low-resource languages, since pre-trained BERT models do not exist for these languages. Note that we only perturbed *one single word* in the attack, therefore, the translation of the rest of the sentence should still generate a reasonable similarity/quality. That is, the measured semantic deviations of the translated texts were all caused by one malicious word in the source. As shown in the table, the translated texts of the adversarial inputs always demonstrate significant deviation from the benign translations. The attack effectiveness is

Table 2. Attack effectiveness for different target languages

Lang.	chrF ↓	BLEU ↓	Rouge-2 ↓	Rouge-L ↓	CIDEr ↓	ADA ↓	BERT ↓
en→fr	70.7	54.45	0.521	0.680	4.021	0.804	0.809
en→de	60.0	38.60	0.339	0.539	3.369	0.694	0.779
en→it	64.1	44.05	0.390	0.574	3.689	0.764	0.781
en→es	66.2	47.38	0.444	0.614	3.011	0.781	0.833
en→ga	67.6	53.50	0.515	0.662	4.678	0.734	-
en→af	72.0	59.48	0.586	0.759	4.934	0.795	-
en→sw	69.0	53.72	0.465	0.654	3.801	0.797	-
en→so	61.8	43.24	0.403	0.581	3.358	0.804	-
en→tr	60.0	42.57	0.346	0.524	3.165	0.717	0.752

shown to be mostly consistent across all destination languages. This is because the source text was compromised (from D_s to D'_s); as long as the MT faithfully translates the input D'_s , the output D'_t will significantly deviate from D_t .

We further present the attack performance for non-English source languages, i.e., attacking the translation from French, German, Italian, and Spanish. The source data are obtained using Google Translate from the English sentences of the corpus used in the English-German experiments (Table 1). We present the experimental results (attack effectiveness measures) in Tables 6 to 10.

4.3 Other Design Choices and Adaptations

4.3.1 Searching the Replacement Word with a Smaller Local LLM

For users unwilling to pay per-use LLM-as-a-service or are concerned with the security of information sent over the internet, a smaller, local model could be used to search for the replacement word (replacing ChatGPT in Step 2 in Fig. 2). However, due to fewer parameters, tighter attention, and less long-form training, small LLMs struggle with longer prompts.

To accommodate the constraints, we increased prompt granularity by decoupling the long prompt into two: (1) to classify the word; (2) to generate a replacement in the returned category.

We evaluate four LLM backends of varying sizes for replacement word generation. As shown in Table 3, we can observe the following: (1) Latency ranged from 0.264 to 45.25 seconds per word on an Apple M1 Max 64G laptop. (2) Nemotron Nano (4B) produces the longest outputs on average, indicating poor instruction-following—it frequently generates reasoning chains or multi-word responses instead of a single replacement word, as required by our prompt. (3) Llama 70B achieves mostly clean single-word outputs but at a significantly higher latency (6.33s per prompt). (4) Llama 8B and Qwen 14B both produce near-clean outputs with substantially lower latency, making them highly practical. Qwen 14B achieves the highest output diversity (42.1%), while Llama 8B offers the fastest response time. These results confirm that model size alone does not determine replacement quality, and that a smaller model with strong instruction-following capability is preferable for this task. Last, we also observed that local models tend to generate replacements in limited diversity, e.g., when asked to perturb the

Table 3. Comparison of LLM backends for replacement word search on 20,000 SQuAD samples. Avg. output length (tokens) measures instruction-following quality. Values reported as ADA-selected target/BERT-selected target.

Model	Para	Time(s)	Out. Len.	Model	Para	Time(s)	Out. Len.
Nemotron Nano	4B	9.19/7.00	3.54/2.11	Llama 3.1	8B	0.48/0.48	1.16/1.09
Qwen2.5 Coder	14B	0.93/0.97	1.10/1.05	Llama 3.3	70B	6.33/5.23	1.10/1.08

Table 4. Attack effectiveness when LLaMa-8B is used to select the replacement word.

Lang.	chrF↓	BLEU↓	Rouge-2↓	Rouge-L↓	CIDEr↓	ADA↓	BERT↓
en→fr	71.1	52.35	0.529	0.702	4.164	0.755	0.818
en→de	69.7	48.78	0.465	0.644	3.365	0.787	0.860
en→it	68.1	49.13	0.466	0.659	3.574	0.702	0.817
en→es	71.2	49.11	0.461	0.646	3.587	0.791	0.858

word “patient” three times, GPT-4 generated “predator”, “violent”, and “agent”, respectively, while Llama 3.3 (70B) returned “impatient” three times.

Attack effectiveness of Llama 70B on four destination languages is shown in Table 4. Most of the semantic deviations of the translated sentences are slightly larger than GPT-4 attacks (Table 2), which indicates that Llama’s choices of replacements are slightly less adversarial than the GPT-generated words.

We conclude that mid-sized LLMs could be employed to locally search for the replacement words. Local LLMs with 14B parameters can generate target words real-time under a high-end personal laptop, while LLMs with 70B parameters can work real-time on desktops with advanced consumer GPUs.

4.3.2 Attacking Other ASR and E2E Speech Translation Models

The attack algorithms described in Section 3 do not depend on any ASR/MT model. To further validate the generalizability of our attack pipeline beyond DeepSpeech, we evaluate it on OpenAI’s Whisper [27], a state-of-the-art transformer-based ASR and end-to-end translation model.

• **Attacking Whisper’s Multilingual ASR Module.** Whisper’s ASR module employs a transformer encoder-decoder architecture with global self-attention, so that the prediction of each output token depends on the entire input sequence instead of a local frame. Therefore, adversarial perturbations must be applied to the entire audio instead of a targeted segment. We followed [23] to modify the noise injection module in our attack, and then evaluated it on 1,000 samples each for English and French as the source. We achieved an average attack success rate of 93.2% and an average attack time of 91 seconds per sample.

• **Attacking Whisper’s End-to-end Speech Translation Model.** Recently, end-to-end (E2E) MT systems have emerged as a new speech translation pipeline [33]. We evaluated our attack against Whisper’s E2E speech translation [27], which performs ASR in any language and translates into English text. The E2E translation directly maps continuous acoustic features to target-language text without an explicit intermediate source-language transcription. Therefore, instead of perturbing the ASR model to flip the target word in the source language (as we have done in attacking Whisper’s ASR), we directly flip the text output

Table 5. Attack effectiveness with MS Azure Translation (top), two LLMs: LLaMa 3.1 8B (middle), and ChatGPT (bottom) as the machine translation module.

Lang.	chrF↓	BLEU↓	Rouge-2↓	Rouge-L↓	CIDEr↓	ADA↓	BERT↓
en→fr	72.4	54.23	0.530	0.694	2.377	0.764	0.866
en→de	67.1	53.84	0.513	0.687	2.473	0.723	0.878
en→it	68.6	53.68	0.492	0.668	1.914	0.607	0.851
en→es	69.3	56.78	0.539	0.716	2.137	0.647	0.881
en→fr	56.6	33.2	0.462	0.603	2.897	0.915	0.853
en→de	55.0	30.68	0.401	0.564	2.531	0.919	0.817
en→it	53.5	28.73	0.361	0.542	2.462	0.919	0.848
en→es	58.5	34.16	0.454	0.614	2.984	0.913	0.858
en→fr	72.1	49.78	0.636	0.760	2.284	0.942	0.926
en→de	70.1	49.01	0.589	0.748	1.506	0.947	0.919
en→it	70.0	48.85	0.560	0.732	1.246	0.944	0.915
en→es	70.6	50.26	0.601	0.745	1.281	0.945	0.931

in the target language in the E2E translation model. With 100 iterations, the attack success rate is 46% and the average NMSE between the benign and altered audio is 2.69×10^3 . Whisper’s higher robustness made attacks harder—more iterations, lower ASR, and higher noise. However, if we use a full English sentence as our attack goal (instead of localized perturbations), we achieve 95%+ attack success rate with an average time of 42.37 seconds per sentence (10-batching).

4.3.3 Adaptations on other Machine Translation Systems

As shown in Figure 1, the speech recognition module is the only employed attack vector. We do not manipulate any other component of the MT pipeline. Therefore, when different translation modules are utilized in the pipeline, the attack is expected to remain effective. To empirically validate the attack against other translators, we evaluated it with three additional MT systems besides Google Translate: (1) Microsoft Azure Translator, a commercially available MT product; (2) LLaMA 3.1 8b, and (3) ChatGPT 4, two LLMs utilized as AI translation tools. As shown in Table 5, across standard metrics, we observed reduced variance across target languages, suggesting that these translators may offer comparatively more stable cross-linguistic performance.

4.4 Ablation Studies

Stop words. Stop words are excluded in target word identification (Sec. 3.2). To evaluate the necessity of this filter, we remove it and re-run word selection on 1,000 SQuAD sentences. The BERT-based method selects a stop word as target in 48.2% of cases, while Ada selects a stop word in only 3.9% of cases, i.e., Ada-embedding is less sensitive to stop words. As an independent reference, an LLM-as-Judge method selects a stop word in only 2.6% of cases. Stop words are semantically vacuous and their “opposites” are hard to define. Hence, they are unsuitable as targets. The results confirm that stop word filtering is essential.

Number of target words. In the attack, we use BERT and Ada-002 to each identify one target word (Section 3.2). To explore the costs/benefits of flipping

multiple words, we invoke Ada and BERT to each identify two target words in 1,000 SQuAD sentences, and then flip both terms in the attack pipeline (en→de). As a result: (1) the attack time doubled; (2) the attack success rate (successfully flipping both words) dropped to 96.5%; (3) NMSE increased to 2.7×10^3 , while STOI dropped to 0.945; (4) in the target language, an average of 2.48 words changed, with mean chrF of 43.76 and BLEU of 17.66; (5) in some cases, flipping two terms broke the grammatical/semantic coherency of the sentence. In summary, identifying and flipping two targets achieves higher semantic distances (attack effectiveness) at the cost of computation and higher noise levels.

Number of replacement words. We search for one replacement word for each target (Section 3.3). To understand the quality of the replacement word, we prompt ChatGPT to generate two phonetically similar candidates for each target and evaluate both on 1,000 SQuAD sentences. Both candidates exhibit nearly identical semantic divergence from the original target word: cosine similarities range narrowly from 0.159 to 0.175, and WordNet Wu-Palmer similarities range from 0.257 to 0.291, with *negligible variation* between the first and second candidates, i.e., the two candidates are not meaningfully differentiated in terms of the semantic contrast. Therefore, we argue that a single candidate is sufficient and that the second provides no substantial additional benefit.

LLM-guided replacement-word search. To evaluate the contribution of LLM-guided phonetically similar word search, we replace it with (1) random replacement words and (2) antonyms.

- With a random replacement word, the attack achieves a success rate of 2.6% in 50 iterations across 500 samples. The reduced success is due to the phonetic gap: a random replacement is phonetically distant from the target, requiring a far larger perturbation to force the ASR to produce the target transcription.
- When an antonym of the target word is selected as the replacement, the attack achieves a targeted success rate of 71.7% in 50 iterations across 947 samples having antonyms in WordNet. We observe two common patterns from the failures: the ASR model either retains a transcription close to the original target word or produces a phonetically unrelated output. The root cause is that antonyms are selected purely on semantic grounds without regard to phonetic proximity, resulting in a large acoustic gap that the adversarial perturbation must bridge. These findings confirm that phonetic similarity between the original and replacement words is essential to attack success.

5 Attack Stealthiness and User Study

5.1 Numerical Assessment

We measure the similarity between the perturbed audio and the original using Normalized Mean Squared Error (NMSE) between two signals, Perceptual Evaluation of Speech Quality (PESQ) [6], Short-Time Objective Intelligibility (STOI) [35], and Extended STOI (ESTOI)[16].

- A lower NMSE indicates the perturbed audio closely resembles the original—the attack is more stealthy. As shown in Table 1, our attack achieves an average

NMSE of 1.4×10^3 , compared to 3.8×10^6 for Baseline (A). That is, our added noises are over 2,500 times smaller, confirming the high stealthiness of our attack.

- PESQ models the human auditory system by comparing the power spectral density of a reference and degraded signal across frequency bands. Our attack achieved a mean PESQ of 2.68. PESQ was designed for globally distributed distortions in telephone degradation, while our attack perturbs only a single word. As a result, PESQ over-penalizes these localized changes, yielding relatively low scores even when the perturbation is largely acceptable to human listeners.

- STOI and ESTOI measure the degree to which the speech signal remains intelligible after degradation. They operate on short-time frames and produce scores in $[0, 1]$. Our attack achieved a mean STOI of 0.961, and a mean ESTOI of 0.918, indicating that the perturbed audio retains near-perfect intelligibility, as listeners can correctly understand the speech in over 96% of cases on average.

5.2 User Study for Attack Stealthiness

We designed an IRB-approved questionnaire to gauge users’ experiences with ASR and translation functions in social media, and their capability in recognizing adversarial audio. We first display the IRB information statement, followed by questions to assess the users’ background: **SQ1**: How much do you send/receive voice messages online? **SQ2**: Have you used the built-in speech-to-text feature in mobile apps when you receive a voice message? **SQ3**: When you see a message in languages other than English or other languages you speak, would you use the built-in “translate” feature? We then provide a brief explanation of noise-based attacks and adversarial noise, followed by five randomly sampled audio clips. Each clip may be benign or adversarial. We asked the participants to carefully listen to the clips and determine if each one was benign or adversarial.

The questionnaire was sent to students in universities in the US. The majority of them are native English speakers. We received 380 annotations from 76 responses. In **SQ1**, 33.9% of the respondents claimed to send voice messages multiple times a week or more often. In **SQ2**, 55.4% of the respondents rated three or more points out of five on reliance on speech-to-text systems when receiving a voice message. In **SQ3**, 58.9% of the respondents rated 3+ points out of 5 regarding reliance on machine translation for information in languages they do not speak. These results suggest that a significant portion of users are highly dependent on automated speech and language processing systems in routine digital interactions. From the results, we observe the following:

1. The overall accuracy (53.4%) was low: 135/194 benign audios and 68/186 adversarial audios were correctly labeled. The results (slightly better than random guesses) suggest that our attack is highly stealthy to human ears.
2. The participants correctly identified 135/194 benign audios, while mistakenly labeled 118/186 adversarial audios as benign. The results reflect a clear trend that participants were more likely to label samples as *benign* regardless of their true label. This bias reflects the subtlety of adversarial perturbations. These findings align with previous expectations that adversarial attacks in the audio domain can be highly effective and remain imperceptible to human listeners.

3. We found no statistically significant correlation between self-reported familiarity with speech-to-text/MT and their ability to detect adversarial manipulations. This suggests that users are equally vulnerable to such attacks, regardless of their prior exposure to speech-based NMT systems.

4. Approximately 1/3 of participants completed within 60 seconds, and another 1/3 within 60 to 120 seconds. As the survey consisted of only five short audio clips (≤ 5 seconds), it is likely that many participants replayed the clips multiple times. We found no statistically significant correlation between the completion time and the identification accuracy. That is, participants who spent more time on the questionnaire do not demonstrate better performance.

The results confirm the practical stealthiness of our attack. Participants demonstrated a limited ability to distinguish adversarial audios from the benign ones. That is, in the real-world attack, even if the users replay the adversarially edited voice messages, they are unlikely to notice the attack.

6 Discussions: Defense

In the literature, three primary defense strategies against adversarial audios have been investigated: (1) *Adversarial training*[47] enhances model robustness by deliberately exposing it to adversarial examples. Carefully crafted perturbations are added to the training data so that the model encounters a wider range of inputs, including malicious ones, while learning. (2) *Input sanitization* approaches pre-process the input audio before it reaches the model. With the observation that adversarial perturbations are usually fragile, this strategy applies signal-processing techniques such as noise filtering and various forms of signal transformation or resampling to suppress/remove the malicious noise while preserving the essential features of the legitimate audio content. (3) *phoneme-level anomaly detection* identifies audio segments whose acoustic characteristics deviate from natural speech patterns. It could serve as a lightweight preprocessing to detect the localized perturbations introduced in our attack. However, the effectiveness of these existing strategies against the specific attack surface explored in this work, namely, semantically targeted word-level substitutions combined with localized audio perturbation, remains an open question. Designing and evaluating defense mechanisms remain an important direction for future work.

7 Related Works

Adversarial Examples have been studied in image, audio, and text processing. In CV, carefully crafted perturbations cause models to misclassify input images with high confidence [13, 8, 20]. Similarly, audio attacks introduce imperceptible perturbations on waveforms to deceive models [43, 41, 15]. They have been adopted against speech recognition systems, where altered audio causes misinterpretation of spoken commands or transcription errors [9, 4, 26, 2, 44, 10].

NMT Attacks published in the past five years focused exclusively on text-to-text systems. [37] presented attacks against NMT systems using targeted flips,

malicious nonsense, untargeted universal trigger, and universal suffix dropper. All four attacks target specifically the text space of ML systems. [40] targeted the learning phase of the NMT system. It develops fake websites containing incorrect parallel corpora to poison NMT systems. [38] utilized a simplified approach to poison monolingual data by adding problematic adjective clips of text into legitimate sentences. [30] introduced an attack based on an emotion classifier that aims to change the emotional class of the target word in a sentence to switch the translated sentence to a different emotion. [46] introduced a new greedy-based attack by calculating results based on a pre-trained encoding module that approaches the one used in the black-box NMT system. They managed to maximize the difference of the encoding module to maximally impact NMT. **Speech Translation Attacks** have only begun to attract attention recently. Liu et al. [18] proposed an attack framework that shares a similar attack domain with our approach, particularly in leveraging adversarial perturbations on speech inputs to manipulate downstream translation outputs. However, the objectives and threat models of the two attacks differ substantially. Consistent with earlier adversarial studies, Liu et al. focused on inducing the system to generate a fixed replacement sentence that is unrelated to the context of the ongoing conversation, e.g., the phrase “are you insane?” is used to replace the target sentence, regardless of its original content. Such attacks demonstrate the vulnerability of speech translation systems to targeted manipulation, but do not consider semantic consistency with the original input.

In contrast, our attack conducts a fine-grained analysis of each source sentence and constructs a customized target translation that preserves the semantic context of the original sentence to the maximum, while introducing *controlled and meaningful deviations*. Instead of forcing a semantically unrelated output, our method aims to subtly alter the translated content in a way that is stealthy (not alerting the communicators) yet impactful, highlighting a more realistic and practically concerning threat model against speech translation systems.

8 Conclusion

We present a complete attack pipeline of adversarial example attacks against speech-based MT systems. We develop an end-to-end attack framework to demonstrate that such systems are vulnerable to carefully crafted perturbations that can significantly alter translation outputs without alerting end users. Experimental results demonstrate the feasibility and impact of the attacks, while a user study confirms the stealthiness of the manipulations. Last, we acknowledge that the proposed attack works on cascade MT models only. Our exploration into E2E MT is still preliminary, which will be continued in our future work.

Acknowledgments. This paper was supported in part by US NSF IIS-2014552, DGE-1565570, NSF EPSCoR Research Fellows award OIA-2531881, and the Ripple University Blockchain Research Initiative. We thank the anonymous reviewers for their valuable comments and suggestions.

References

1. gTTS: Google text-to-speech, version 2.5.4. <https://pypi.org/project/gTTS/> (2024), python package; Released: Nov 10, 2024; Last accessed: Sept 2025
2. Abdullah, H., Warren, K., Bindschaedler, V., Papernot, N., Traynor, P.: Sok: The faults in our asrs: An overview of attacks against automatic speech recognition and speaker identification systems. In: IEEE Symp. on Security & Privacy (2021)
3. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
4. Alzantot, M., Balaji, B., Srivastava, M.: Did you hear that? adversarial examples against automatic speech recognition. arXiv preprint arXiv:1801.00554 (2018)
5. Barrault, L., Chung, Y.A., Meglioli, M.C., Dale, D., Dong, N., Duquenne, P.A., Elshahar, H., Gong, H., Heffernan, K., Hoffman, J., et al.: Seamlessm4t: Massively multilingual & multimodal machine translation. arXiv:2308.11596 (2023)
6. Beerends, J.G., Hekstra, A.P., Rix, A.W., Hollier, M.P., et al.: Perceptual evaluation of speech quality (pesq): The new itu standard for end-to-end speech quality assessment part ii-psychoacoustic model. JAES **50**(10) (2002)
7. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., et al.: Language models are few-shot learners. NeurIPS (2020)
8. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: IEEE Symposium on Security and Privacy (2017)
9. Carlini, N., Wagner, D.: Audio adversarial examples: Targeted attacks on speech-to-text. In: 2018 IEEE security and privacy workshops (SPW) (2018)
10. Cheng, P., Wang, Y., Huang, P., Ba, Z., Lin, X., Lin, F., Lu, L., Ren, K.: Alif: Low-cost adversarial audio attacks on black-box speech platforms using linguistic features. In: IEEE Symposium on Security and Privacy (2024)
11. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. CoRR **abs/1911.02116** (2019)
12. Dabre, R., Chu, C., Kunchukuttan, A.: A survey of multilingual neural machine translation. ACM Computing Surveys (CSUR) **53**(5) (2020)
13. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
14. Grusky, M.: Rogue scores. In: Annual Meeting of the ACL (Jul 2023)
15. Hussain, S., Neekhara, P., Dubnov, S., McAuley, J., Koushanfar, F.: {WaveGuard}: Understanding and mitigating audio adversarial examples. In: USENIX Sec (2021)
16. Jensen, J., Taal, C.H.: An algorithm for predicting the intelligibility of speech masked by modulated noise maskers. IEEE/ACM TASLPRO **24**(11) (2016)
17. Koehn, P.: Statistical machine translation. Cambridge University Press (2009)
18. Liu, C., Wu, H., Yang, X., Zhang, K., Wu, C., Zhang, W., Yu, N., Zhang, T., Guo, Q., Zhang, J.: Exploiting vulnerabilities in speech translation systems through targeted adversarial attacks. arXiv preprint arXiv:2503.00957 (2025)
19. Lopez, A.: Statistical machine translation. ACM Computing Surveys **40**(3) (2008)
20. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR (2018)
21. Maruf, S., Saleh, F., Haffari, G.: A survey on document-level neural machine translation: Methods and evaluation. ACM Computing Surveys (CSUR) **54**(2) (2021)
22. Miller, G.A.: Wordnet: a lexical database for english. Comm. of ACM **38**(11) (1995)

23. Olivier, R., Raj, B.: There is more than one kind of robustness: Fooling whisper with adversarial examples. In: *Proc. Interspeech 2023*. pp. 4394–4398 (2023)
24. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: *Proceedings of the ACL* (2002)
25. Popović, M.: chrF: character n-gram f-score for automatic mt evaluation. In: *Proceedings of the tenth workshop on statistical machine translation* (2015)
26. Qin, Y., Carlini, N., Cottrell, G., Goodfellow, I., Raffel, C.: Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In: *ICML* (2019)
27. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision. In: *ICML* (2023)
28. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics (2016)
29. Rivera-Trigueros, I.: Machine translation systems and quality assessment: a systematic review. *Language Resources and Evaluation* **56**(2) (2022)
30. Sadrizadeh, S., Dolamic, L., Frossard, P.: A classification-guided approach for adversarial attacks against neural machine translation. In: *EACL* (Mar 2024)
31. Shiwen, Y., Xiaojing, B.: Rule-based machine translation. In: *Routledge encyclopedia of translation technology*. Routledge (2014)
32. Slocum, J.: A survey of machine translation: Its history, current status and future prospects. *Computational linguistics* **11**(1) (1985)
33. Sperber, M., Paulik, M.: Speech translation and the end-to-end promise: Taking stock of where we are. In: *Proceedings of the 58th Annual Meeting of ACL* (2020)
34. Stahlberg, F.: Neural machine translation: A review. *Journal of Artificial Intelligence Research* **69** (2020)
35. Taal, C.H., Hendriks, R.C., Heusdens, R., Jensen, J.: A short-time objective intelligibility measure for time-frequency weighted noisy speech. In: *ICASSP* (2010)
36. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: Cider: Consensus-based image description evaluation. In: *IEEE CVPR* (2015)
37. Wallace, E., Stern, M., Song, D.: Imitation attacks and defenses for black-box machine translation systems. In: *EMNLP* (Nov 2020)
38. Wang, J., Xu, C., Guzmán, F., El-Kishky, A., Tang, Y., Rubinstein, B., Cohn, T.: Putting words into the system’s mouth: A targeted attack on neural machine translation using monolingual data poisoning. In: *Findings of the ACL* (2021)
39. Weiss, R.J., Chorowski, J., Jaitly, N., Wu, Y., Chen, Z.: Sequence-to-sequence models can directly translate foreign speech. In: *Proc. Interspeech* (2017)
40. Xu, C., Wang, J., Tang, Y., Guzmán, F., Rubinstein, B.L., Cohn, T.: A targeted attack on black-box neural machine translation with parallel data poisoning. In: *Proceedings of the web conference 2021* (2021)
41. Yakura, H., Sakuma, J.: Robust audio adversarial example for a physical attack. In: *Proceedings of the Twenty-Eighth IJCAI*. pp. 5334–5341. *IJCAI* (2019)
42. Yang, S., Wang, Y., Chu, X.: A survey of deep learning techniques for neural machine translation. *arXiv preprint arXiv:2002.07526* (2020)
43. Yang, Z., Li, B., Chen, P.Y., Song, D.: Characterizing audio adversarial examples using temporal dependency. In: *ICLR* (2018)
44. Yu, Z., Chang, Y., Zhang, N., Xiao, C.: {SMACK}: Semantically meaningful adversarial audio attack. In: *USENIX security* (2023)
45. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: Evaluating Text Generation with BERT. In: *ICLR* (2020)

46. Zhang, X., Zhang, J., Chen, Z., He, K.: Crafting adversarial examples for neural machine translation. In: ACL-IJCNLP (2021)
47. Zhao, M., Zhang, L., Ye, J., Lu, H., Yin, B., Wang, X.: Adversarial training: A survey. arXiv preprint arXiv:2410.15042 (2024)

A Attack Effectiveness Metrics

(1) Text Embedding Similarity. Sentences are encoded into dense vector representations in pretrained language models. We use the BERTScore and text-embedding-ada-002 and compute semantic deviation as the cosine similarity between the embeddings of the clean and adversarial translations. For low-resource languages without BERT, we use “xlm-roberta-large” as an approximation.

(2) chrF (Character-level F-score). A standard, language-agnostic metric that calculates the overlap of character n-grams between two translations. chrF [25] is sensitive to subtle lexical and grammatical differences that may not be detected by token-based or sentence-level metrics.

(3) BLEU (Bilingual Evaluation Understudy). BLEU [24] is a widely-used metric for evaluating the quality of machine-generated text, especially in machine translation. It calculates how much the model’s output overlaps with reference translations based on matching word n-grams. A higher BLEU score indicates that the output is more fluent and accurate.

(4) ROUGE (Recall-Oriented Understudy for Gisting Evaluation). A family of metrics designed for automatic text summarization evaluation, but also widely applied to MT and other NLP tasks. ROUGE [14] measures the overlap of n-grams, word sequences, and word pairs between a system output and the reference texts. Among its variants, ROUGE-N (n-gram recall), ROUGE-L (longest common subsequence), and ROUGE-S (skip-bigrams) are most popular. We adopt ROUGE as a complement to BLEU, as it emphasizes recall rather than precision, making it particularly effective for detecting missing content.

(5) CIDEr (Consensus-based Image Description Evaluation): Originally proposed for image captioning evaluation, CIDEr computes the similarity between a candidate text and a set of reference texts based on the weighted n-gram matching, where the weights are derived using TF-IDF. We adopted CIDEr-D [36] in our evaluation to capture semantic adequacy and penalize generic or trivial outputs, thereby complementing metrics such as BLEU and ROUGE with a stronger emphasis on content relevance and diversity.

B Additional Details for the Experiments

In Table 1, we used the following prompt to generate the replacement word:

Switch the word to an antonym of the word with similar pronunciation, such as dawn to dusk. Just output the word. If it is a number, output another number with a very different value, such as 10 to 5678. if it is a name, output another name with a similar category but very different expertise, such as Einstein to Picasso.

Table 6. Average chrF for language pairs, smaller values mean more effective attacks.

$\begin{smallmatrix} & to \\ from & \end{smallmatrix}$	fr	de	it	es	en	ga	af	sw	so	tr
fr	-	66.1	69.0	70.5	70.1	72.0	65.4	66.2	60.3	63.9
de	65.6	-	64.5	64.3	68.5	68.0	65.5	64.4	57.3	62.1
it	69.7	64.3	-	69.2	69.6	69.6	64.3	68.4	58.7	64.6
es	69.6	65.4	67.5	-	69.8	69.0	64.7	67.5	58.4	64.4

Table 7. Average BLEU for language pairs; f: translate from; t: translate to; smaller values mean more effective attacks.

$\begin{smallmatrix} & t \\ f & \end{smallmatrix}$	fr	de	it	es	en	ga	af	sw	so	tr
fr	-	43.85	48.87	49.80	56.56	57.19	47.52	45.98	38.52	40.79
de	46.62	-	46.31	45.03	55.58	52.06	52.33	44.86	36.24	40.51
it	50.47	44.42	-	50.99	57.05	55.37	49.08	49.28	35.57	42.62
es	50.13	45.64	49.19	-	56.34	54.57	48.92	48.84	35.48	43.20

Table 8. Average ROUGE-2 for language pairs; f: translate from; t: translate to; smaller values mean more effective attacks.

$\begin{smallmatrix} & t \\ f & \end{smallmatrix}$	fr	de	it	es	en	ga	af	sw	so	tr
fr	-	0.538	0.572	0.625	0.639	0.666	0.584	0.504	0.426	0.499
de	0.573	-	0.514	0.543	0.632	0.614	0.600	0.493	0.406	0.482
it	0.626	0.547	-	0.622	0.653	0.649	0.598	0.539	0.412	0.505
es	0.623	0.562	0.556	-	0.652	0.641	0.587	0.527	0.407	0.518

Table 9. Average ROUGE-L for language pairs; f: translate from; t: translate to; smaller values mean more effective attacks.

$\begin{smallmatrix} & t \\ f & \end{smallmatrix}$	fr	de	it	es	en	ga	af	sw	so	tr
fr	-	0.716	0.720	0.755	0.796	0.767	0.765	0.684	0.608	0.659
de	0.707	-	0.677	0.689	0.777	0.743	0.764	0.672	0.584	0.638
it	0.751	0.712	-	0.747	0.799	0.755	0.771	0.710	0.587	0.656
es	0.747	0.723	0.718	-	0.803	0.752	0.763	0.712	0.592	0.672

Table 10. Average CIDEr for language pairs; f: translate from; t: translate to; smaller values mean more effective attacks.

$\begin{smallmatrix} & t \\ f & \end{smallmatrix}$	fr	de	it	es	en	ga	af	sw	so	tr
fr	-	5.219	5.888	5.920	6.216	6.326	5.786	5.308	4.490	4.735
de	5.926	-	5.367	5.386	6.210	5.911	6.088	5.240	4.264	4.794
it	6.332	5.235	-	6.000	6.294	6.172	5.883	5.612	4.270	4.953
es	6.330	5.325	5.876	-	6.280	6.161	5.841	5.601	4.312	5.064