NoWhisper: Fast Generative Perturbations for Breaking Whisper ASR

July 9, 2026

Lorenzo Marinelli 2043092 Andrea Venditti 2031589

Abstract

NoWhisper studies untargeted adversarial degradation of Whisper ASR. The project started from a slow white-box waveform attack and then moved to the more useful question: can this expensive per-sample optimization be replaced by a fast neural generator? Our best model is a residual 1D U-Net that predicts an additive waveform perturbation in one forward pass, under an L_∞ budget and the same clipping constraints used by the white-box attack. A key result is that the strongest generator was not obtained by simply imitating a white-box teacher. Teacher-oriented training was implemented and tested, but the best results came from disabling teacher imitation and training the generator directly against Whisper's teacher-forced reference loss. On Whisper Small, the final $\epsilon = 0.0075$ generator raises WER from 0.047 to 3.825 on a held-out test subset, with 99% attack success and mean SNR 20.79 dB. The same generator also transfers strongly without retraining: Whisper Medium goes from WER 0.033 to 2.242, and Whisper Large-v3 goes from 0.023 to 2.817. This makes Whisper Small a useful and cheap surrogate for producing perturbations that remain adversarial to larger Whisper models.

1. Introduction

Whisper is a sequence-to-sequence ASR system trained on large-scale weakly supervised speech data (Radford et al., 2023). In ordinary conditions it is very strong: background noise, imperfect recordings, and speaker variation often do not break it. This robustness, however, is not the same as adversarial robustness. Neural models can be highly sensitive to small, optimized perturbations (Goodfellow et al., 2015), and speech-to-text systems have already been shown to be vulnerable to waveform-level adversarial examples (Carlini & Wagner, 2018; Olivier & Raj, 2022).

NoWhisper focuses on untargeted degradation. We do not try to force Whisper to output a chosen malicious sentence. Instead, we try to make the model stop transcribing the original speech correctly. This is still a serious failure mode: ASR is often used inside indexing, accessibility tools, moderation, meeting summaries, and automation pipelines. A wrong transcript can silently become wrong downstream information.

The project has two parts. The first part is a slow white-box attack. For each audio clip, it optimizes a separate perturbation by differentiating through Whisper's reference loss. This is useful because it gives a direct adversarial objective, but it is too expensive to be the final solution. The second part is a generator attack: train a neural network once, then use it to produce a perturbation for new audio in a single forward pass, following the motivation of fast generative audio attacks (Xie et al., 2020).

The most important design lesson was that teacher imitation was not enough. We implemented teacher-oriented generator training, where the generator tries to match perturbations produced by the slow attack, in the spirit of distillation (Hinton et al., 2015). In practice, these teacher deltas were too sample-specific. The best generator was obtained when teacher imitation was turned off and the model was trained directly to increase Whisper's loss on the original transcript.

The second important result is transfer. Training through Whisper Small is much cheaper than training through Whisper Medium or Large-v3. Nevertheless, the perturbations learned on Small remain highly damaging on the larger models. This suggests that the generator is not only exploiting a private weakness of one checkpoint, but is hitting sensitivities shared across the Whisper family. The code for the project is available at `https://github.com/Reewd/NoWhisper`. We also provide an online demonstration page with white-box attack examples, audio comparisons, and resulting transcripts at `https://dlai.avenditti.com/`. This demo is useful because it makes the failure mode less abstract: the listener can

Email: Lorenzo Marinelli <marinelli.2043092@studenti.uniroma1.it>, Andrea Venditti <venditti.2031589@studenti.uniroma1.it>.

Deep Learning and Applied AI 2026, Sapienza University of Rome, 2nd semester a.y. 2025/2026.

compare clean and adversarial audio while also seeing how Whisper's output changes during the attack.

2. Method

The data comes from LibriSpeech clips (Panayotov et al., 2015). Each example contains a waveform and a reference transcript. Audio is loaded as 16 kHz mono waveform data and kept in the valid range $[-1, 1]$. Whisper is accessed through Hugging Face Transformers (Wolf et al., 2020). Clean decoding is first used to establish WER and CER baselines; adversarial performance is then measured against the same original transcript.

White-box attack. For a clean waveform x and transcript y, the white-box attack initializes a perturbation δ at zero and optimizes it with Adam. After each step, the perturbation is projected into the L_∞ budget and the final audio is clipped:

$$x_{\text{adv}} = \text{clip}\left(x + \Pi_{[-\epsilon,\epsilon]}(\delta), -1, 1\right). \tag{1}$$

The attack is untargeted. Its goal is not to produce a specific wrong sentence, but to increase the cross-entropy of the correct transcript under teacher-forced decoding:

$$\min_{\delta} -L_{\text{ref}}(x_{\text{adv}}, y) + \lambda\|\delta\|_2^2. \tag{2}$$

Here L_{ref} is the mean token-level reference loss. The decoder receives shifted reference tokens, so the loss can be computed without calling autoregressive generation. The implementation also supports masking Whisper prefix tokens from the loss while still keeping them as decoder context. This attack gives a strong reference point, but it requires many gradient steps per clip.

Generator attack. The generator replaces per-clip optimization with a feed-forward model G_θ. Given clean audio x, it predicts raw perturbation logits. The implementation converts these logits into a bounded perturbation using $\tanh$, masks padded samples, adds the perturbation to the clean waveform, clips to $[-1, 1]$, and then recomputes the effective delta:

$$x_{\text{gen}} = \text{clip}\left(x + \epsilon \tanh(G_\theta(x)), -1, 1\right). \tag{3}$$

This is exactly the same constraint logic used during evaluation: the reported perturbation is the effective waveform difference after clipping, not just the raw generator proposal.

The best evaluated architecture is `resunet1d`: a residual 1D U-Net with base width 8, depth 4, kernel size 15, and two residual blocks per stage. The residual stages use Conv1D layers, GroupNorm, SiLU activations, and dilations inside the residual units. Encoder–decoder skip connections let the model combine local waveform information with a larger temporal context.

Implemented training objective. The training script supports both teacher-oriented and teacher-free objectives. In its full form, the optimized objective can be written by separating the attack terms from the waveform regularizers:

$$\begin{aligned}\mathcal{J}(\theta) = &- w_{\text{adv}} L_{\text{ref}}^{\text{EOT}}(x_{\text{gen}}, y) + w_{\text{teach}} D_{\text{mask}}(\delta_\theta, \delta_T) \\ &+ w_{\text{stftT}} D_{\text{stft}}(\delta_\theta, \delta_T) + R_{\text{gen}},\end{aligned} \tag{4}$$

$$\begin{aligned}R_{\text{gen}} = &\lambda_{\text{mse}} R_{\text{mse}} + \lambda_{\text{sm}} R_{\text{sm}} + \lambda_{\text{hf}} R_{\text{hf}} \\ &+ \lambda_{\text{le}} R_{\text{le}} + \lambda_{\text{lsnr}} R_{\text{lsnr}} + \lambda_{\text{stft}} R_{\text{stft}}.\end{aligned} \tag{5}$$

Here $\delta_\theta = x_{\text{gen}} - x$ is the effective generated perturbation. The first term is the direct adversarial term. $L_{\text{ref}}^{\text{EOT}}$ is the mean Whisper reference loss, optionally averaged over stochastic audio views when EOT augmentation is enabled. Since the objective is minimized, the negative sign makes the generator increase Whisper's loss on the correct transcript.

$D_{\text{mask}}(\delta_\theta, \delta_T)$ is the optional teacher-delta imitation loss. The teacher delta is $\delta_T = x_T - x$, where x_T is an adversarial waveform produced by the slow white-box attack. The code computes this loss only over valid, unpadded waveform samples, with either Smooth L1 or MSE. It can also normalize both deltas by ϵ so that small-budget imitation errors remain numerically visible. D_{stft} is a separate optional teacher spectral loss: it compares $\log(1 + |\text{STFT}|)$ magnitudes of generated and teacher deltas at FFT sizes 256, 512, and 1024.

The remaining terms are generator regularizers, grouped in R_{gen}. R_{mse} is the mean squared perturbation amplitude, so it directly discourages large average energy. R_{sm} is the mean square of first differences, which discourages rapid sample-to-sample jumps. R_{hf} is the mean square of second differences, penalizing sharper high-frequency curvature. R_{le} is a clean-relative local energy penalty: the code computes local clean power and local delta power in a sliding window, then penalizes the ratio between them with a masking floor. R_{lsnr} uses the same ratio but only penalizes violations above a target local-SNR threshold. Finally, R_{stft} penalizes perturbation STFT energy where the clean speech STFT has little energy, which is a simple masking-inspired regularizer related to perceptual audio attack work (Schönherr et al., 2018; Qin et al., 2019).

The best reported generator is teacher-free in the important sense: teacher imitation is disabled with $w_{\text{teach}} = 0$, and the spectral teacher imitation term is also disabled. The generator is therefore not trained to copy the white-box perturbation. It is trained to find its own perturbation by directly increasing Whisper's reference loss while paying waveform regularization penalties.

Table 1. Generator results on Whisper Small. WER/CER compare clean and adversarial decoding. SNR is reported in dB.

Run	ϵ	WER	CER	SNR
R-U-Net 20ep	.005	.036→2.165	.012→1.616	23.37
Strong mask	.005	.036→.064	.012→.029	23.64
R-U-Net	.0075	.036→3.685	.012→2.918	20.06

Table 2. Cross-model transfer of the same generator. The perturbation was trained on Whisper Small and evaluated without retraining. Each row uses 100 test samples.

Eval model	WER	CER	Success	SNR
Small	.047→3.825	.020→3.028	99%	20.79
Medium	.033→2.242	.014→1.860	98%	20.79
Large-v3	.023→2.817	.009→2.248	96%	20.79

3. Results

The main generator experiments use Whisper Small for training, batch size 4, learning rate 10^{-4}, and residual U-Net generators. WER and CER are measured by decoding the adversarial audio and comparing it with the original reference transcript.

Table 1 shows the tradeoff between strength and masking pressure. With $\epsilon = .005$, the residual U-Net already produces a strong attack while keeping SNR above 23 dB. Increasing the budget to $\epsilon = .0075$ makes the attack much stronger and pushes WER far above 3.0. The strong-mask run is the useful negative result: it keeps SNR high, but the WER barely changes. This means the attack depends on using enough of the waveform budget in places where Whisper is sensitive; simply adding regularized low-energy perturbations is not sufficient.

The most important experiment is cross-model transfer. The same $\epsilon = .0075$ generator trained on Whisper Small is evaluated on Whisper Small, Medium, and Large-v3 without retraining.

Table 2 is the strongest result of the project. Whisper Medium and Large-v3 are better clean recognizers than Whisper Small, but the Small-trained perturbation still breaks them. Large-v3 starts from the best clean WER, 0.023, and still rises to 2.817. The success rate also remains high across models: 99% on Small, 98% on Medium, and 96% on Large-v3.

This is important because it changes the practical cost of the attack. Training directly through Large-v3 would be much heavier in memory and computation. Instead, Whisper Small can be used as a cheaper surrogate model, and the resulting generator still transfers to larger Whisper variants. The perturbation is therefore not just overfitting one small checkpoint; it appears to exploit model-family sensitivities shared across sizes.

Qualitatively, the adversarial audio often remains understandable to a human listener, while Whisper produces unrelated words, long insertions, or unstable transcripts. The large WER values are not only caused by deleting speech. In many cases Whisper inserts extra text, which is why WER can exceed 1.0 by a large margin. Concrete white-box attack examples, audio comparisons, and resulting transcripts are available on the project demo page: https://dlai.avenditti.com/.

4. Discussion

The main conclusion is that fast generator attacks work. The white-box attack is useful for understanding the objective, but it is too slow because it solves a separate optimization problem for every waveform. The residual U-Net amortizes that process: after training, it predicts perturbations immediately.

The teacher experiments were useful, but mostly because they showed what not to rely on. Copying white-box deltas sounds natural, and the code supports both waveform teacher imitation and spectral teacher imitation. But the strongest results came when the generator was not forced to match the teacher perturbation. Direct adversarial training gave the generator more freedom: it only had to make Whisper lose the transcript while satisfying the waveform constraints.

The main limitation is perceptual quality. An SNR around 20–23 dB is not silent, and stronger masking can make the attack collapse. This is the central tradeoff in NoWhisper: the perturbation must be constrained enough to preserve the speech for humans, but not so constrained that it becomes harmless to Whisper.

Overall, NoWhisper shows that a compact residual generator trained on Whisper Small can produce fast, high-impact adversarial perturbations. The transfer results are especially strong: the same perturbation generator remains effective on Whisper Medium and Large-v3 without retraining. In this setting, a smaller model is not just a cheap training target; it is a useful surrogate for attacking larger models in the same family.

5. Statement on the use of AI

We used ChatGPT to brainstorm project directions, find and summarize related work, and reason about Whisper gradients, white-box losses, generator training, and transfer experiments. We used Codex with GPT 5.5 variants to speed up implementation, refactoring, testing, experiment scripts, and report proofreading. The project design, experiments, result selection, interpretation, and final report content were reviewed and controlled by us.

References

Carlini, N. and Wagner, D. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 1–7. IEEE, 2018.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

Olivier, R. and Raj, B. There is more than one kind of robustness: Fooling Whisper with adversarial examples, 2022.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. LibriSpeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210. IEEE, 2015.

Qin, Y., Carlini, N., Cottrell, G., Goodfellow, I., and Raffel, C. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5231–5240. PMLR, 2019.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28492–28518. PMLR, 2023.

Schönherr, L., Kohls, K., Zeiler, S., Holz, T., and Kolossa, D. Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding, 2018.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45. Association for Computational Linguistics, 2020.

Xie, Y., Li, Z., Shi, C., Liu, J., Chen, Y., and Yuan, B. Enabling fast and universal audio adversarial attack using generative model, 2020.