ROBUSTNESS AFTER SPECIALIZATION: INSTALLING ADVERSARIAL ROBUSTNESS IN SPEECH FOUNDATION MODEL DESCENDANTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Speech foundation models are often specialized into task-specific descendants before adversarial robustness becomes a deployment requirement. We study *post-hoc robustness installation*: can robustness learned once at a foundation backbone be installed into an already fine-tuned descendant without downstream labels, the original fine-tuning pipeline, per-descendant adversarial training, or additional inference-time computation? From unlabeled source audio, we learn a task-agnostic backbone update Δ_{rob}. It reduces adversarial representation drift while explicitly minimizing clean-representation changes. For each descendant, we add the same update at a fixed, untuned scale. We then freeze the patched encoder and realign its linear readout on clean unlabeled audio. We evaluate three backbones on five speech tasks, producing 15 descendants through full encoder fine-tuning. With one shared update per backbone, seven-attack robust accuracy improves by 48.7 percentage points on average across the 12 classification descendants. The worst-task clean-accuracy loss is 7.3 percentage points. The method recovers 79.5% of the task-specific classification AT gain. Evaluation attacks the exact installed model, including with adaptive white-box attacks. Our analysis measures transferred representation stability and relates the remaining adversarial drift to downstream decision margins via an attack-relative sufficient criterion.

1 INTRODUCTION

Speech foundation models such as Wav2Vec2, WavLM, and SpeechT5 (Baevski et al., 2020; Chen et al., 2022; Ao et al., 2022) are often fine-tuned into task-specific *descendants* before adversarial robustness becomes a deployment requirement. When this requirement arises, the defender may have the descendant's weights and clean unlabeled audio but lack its labels, fine-tuning data, or training pipeline. These resources may be proprietary, license-restricted, or no longer retained. Their absence complicates supervised task-specific adversarial training (AT) (Madry et al., 2018), which requires downstream labels and separate robust training for each descendant. Robust-first methods face a mismatch: they require a robust backbone before specialization. Inference-time defenses offer another route, but leave the underlying model vulnerability unchanged and add inference-time computation, which our deployment setting excludes. We therefore study how to add robustness to existing descendants under these constraints while retaining their task-specific capabilities.

Speech makes this setting particularly relevant because downstream specialization often modifies the pretrained encoder itself. In many CLIP-based vision-language systems, the visual encoder is frozen, enabling replacement with a robust counterpart without retraining the downstream model (Schlarmann et al., 2024). In speech, fine-tuning the pretrained encoder is common; wav2vec 2.0 (Baevski et al., 2020), for example, is routinely fine-tuned for downstream tasks. Replacing these encoders would discard their specialization, so robustness must be added without overwriting it. Meeting this requirement with one shared update is demanding because the tasks need different information. Speaker identification relies on speaker characteristics that speech recognition aims to ignore. Emotion recognition uses paralinguistic variation that both may disregard. Yet all descendants process raw waveforms under the same minimum-SNR threat model. Adversarial vulnerabilities inherited from a common pretrained speech encoder can also persist after independent downstream fine-tuning (Wu et al., 2022; Olivier et al., 2022). This shared source of vulnerability motivates

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.

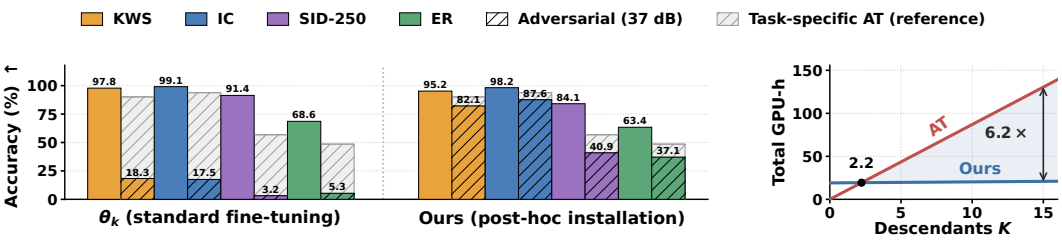

Figure 1: **Installed robustness after specialization.** *Left:* WavLM-large (primary backbone) classification descendants; solid bars show clean accuracy, hatched bars show pooled seven-attack robust accuracy, and gray hatched extensions show the task-specific AT robust-accuracy reference. *Right:* mean compute across the evaluated backbones and tasks for K descendants: task-specific AT averages 8.7 GPU-h per descendant, while installation averages 19.1 GPU-h once per backbone plus 0.12 GPU-h per descendant, breaking even at $K \approx 2.2$. Table 2 reports ASR and other backbones.

testing a common robustness update across tasks. Speech therefore provides a concrete instance of a broader post-hoc installation problem that can arise whenever downstream specialization modifies a shared pretrained encoder. We therefore ask:

Can robustness learned once at a speech foundation backbone be installed in already fine-tuned descendants without downstream labels, the original fine-tuning pipeline, or per-descendant AT?

Installing robustness after specialization creates two obstacles. Write the descendant's encoder weights as $\theta_k = \theta_b + \tau_k$, where θ_b denotes the pretrained backbone weights and τ_k its specialization update. First, replacing θ_k with a robust backbone θ_r discards τ_k. Adding a robustness update retains τ_k in parameter space, but its ability to reduce adversarial representation drift, defined as the feature change caused by an attack, may not transfer from θ_b to θ_k. Second, the update may shift clean representations relative to the existing readout, potentially reducing their decision margins. The decision margin is the distance to the nearest decision boundary, and a correct prediction remains unchanged when adversarial drift stays below this margin. We therefore examine both drift reduction and how the residual drift compares with the decision margin.

We address these obstacles in two stages. **Stage 1** learns a task-agnostic robustness update Δ_{rob} once per backbone from unlabeled source audio to suppress adversarial drift. **Stage 2** installs the learned update in each descendant at the same fixed, untuned scale, then freezes the patched encoder and realigns only its linear readout using clean unlabeled downstream audio. The descendant's original predictions provide supervision, and a geometric-margin objective encourages larger margins. Neither stage uses downstream labels or downstream adversarial examples, and installation requires no original fine-tuning pipeline. The resulting performance and compute are summarized in Fig. 1.

Contributions. **(i) Robustness after specialization:** we formulate a post-hoc setting in which robustness is added after encoder specialization, without downstream labels, the original fine-tuning pipeline, or a task-matched robust model. **(ii) Two-stage label-free installation:** we learn one task-agnostic update per backbone, retain the specialization update through additive composition, and realign only the readout using clean unlabeled audio. **(iii) Drift and margin analysis:** we measure drift reduction and derive a sufficient margin condition for robustness within the evaluated attack pool. **(iv) Amortized robustness:** across three backbones and five tasks, one robustness update per backbone improves seven-attack robust accuracy by 48.7 percentage points on average across 12 classification descendants, with a worst-task clean-accuracy loss of 7.3 points. We recover 79.5% of the task-specific AT gain across the classification tasks. At $K = 15$ descendants per backbone, the amortized cost averages 16% of task-specific AT per model across backbones and tasks.

2 Related Work

Robust foundation models and robust transfer. Foundation-level robustness is usually introduced *before* downstream specialization. TeCoA (Mao et al., 2023) adversarially adapts CLIP with text supervision; FARE/Robust CLIP (Schlarmann et al., 2024) learns robust visual embeddings without text supervision; and LORE (Khodabandeh et al., 2025) uses constrained optimization to

explicitly control clean-embedding deviation during robustification. Robust transfer learning then asks whether such robustness survives later adaptation, finding that freezing can preserve it while end-to-end fine-tuning can erode it (Shafahi et al., 2020). We start after that adaptation has already occurred. Encoder replacement and robust-first adaptation are therefore informative references but operate under a different order of operations; Appendix A discusses these distinctions in more detail, including an objective-level comparison.

Weight-space transfer of learned updates. Parameter differences induced by fine-tuning can encode reusable behavior, motivating task arithmetic, weight interpolation, model soups, and interference-aware merging (Ilharco et al., 2023; 2022; Wortsman et al., 2022; Yadav et al., 2023). Robust Weight Signatures (RWS) (Cai et al., 2023) derive corruption-specific patches from pairs of standard and corruption-robust models, while Dong et al. (2026) merge task vectors from naturally and adversarially fine-tuned vision-language models. These results motivate update reuse, but do not establish that an upstream adversarial-robustness update remains effective after heterogeneous downstream specialization. Our update is learned without knowledge of downstream tasks or a task-matched robust model, then installed into already specialized descendants and evaluated against adaptive waveform attacks. We therefore measure whether it still reduces adversarial representation drift in each descendant, rather than assuming that parameter-space composition preserves its functional effect.

Adversarial robustness in speech. Waveform attacks can induce speech-recognition failures (Carlini & Wagner, 2018; Schönherr et al., 2019; Qin et al., 2019). In self-supervised speech, adversarial vulnerability inherited from shared pretraining can persist across downstream tasks and independently fine-tuned models (Wu et al., 2022; Olivier et al., 2022). Existing defenses include task-specific AT and inference-time defenses such as smoothing and purification (Olivier & Raj, 2021; Wu et al., 2023). The persistence of vulnerability motivates the converse possibility that robustness may also transfer across descendants, but existing work does not address post-hoc installation. We instead modify already-specialized models using robustness learned upstream and only clean unlabeled audio downstream.

Operational distinction. The closest alternatives do not satisfy all conditions of our setting. Robust-first methods act before specialization; replacing the encoder would discard encoder-side specialization; task-vector robustness methods require a task-matched robust model or vector; and task-specific AT requires downstream supervision and a separate robust-training run. Our installation instead preserves specialization through additive composition and uses only the available model weights and clean unlabeled downstream audio. Appendix B tabulates these access assumptions.

3 METHOD

Design logic. The two obstacles in Section 1 motivate our two-stage design (Fig. 2). Stage 1 learns a task-agnostic update once per backbone. Stage 2 installs it additively in each descendant, retaining the specialization update in parameter space, then freezes the patched encoder and realigns only its readout using clean unlabeled audio.

3.1 SETTING AND THREAT MODEL

Let θ_b denote the pretrained encoder weights. Downstream fine-tuning produces a task-specific descendant $\theta_k = \theta_b + \tau_k$, where τ_k is the specialization-induced parameter change. Let $h_\theta(x)$ denote the final-layer encoder representation, mean-pooled for utterance classification and retained frame-wise for CTC. The descendant's original linear readout is $g_0(z) = W_0 z + b_0$. We learn the robustness update from unlabeled source audio $\mathcal{D}_s$ and realign the readout using clean unlabeled in-distribution audio $\mathcal{D}_k^{\text{cal}}$ held out from downstream specialization. To normalize Stage 1 across backbones, we measure representation distances in units of C, the mean ℓ_2 norm of the final-layer frame representations on held-out source audio. The $1/C^2$ factor in Eq. 2 scales both terms equally, leaving the minimizer unchanged while normalizing the objective scale across backbones.

For waveform $x \in [-1, 1]^n$, define the SNR-constrained threat set

$$\Delta_S(x) = \left\{ \delta : \ \|\delta\|_2 \leq \|x\|_2 10^{-S/20}, \ x_i + \delta_i \in [-1, 1] \ \forall i \right\}. \tag{1}$$

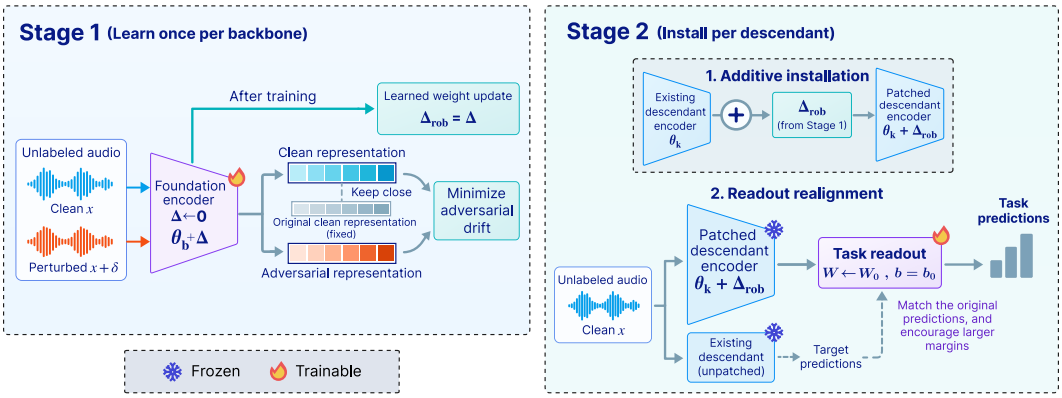

Figure 2: **Learn once, install per descendant.** Starting from pretrained encoder θ_b, downstream fine-tuning produces descendants $\theta_k = \theta_b + \tau_k$. **Stage 1:** unlabeled source audio is used once per backbone to learn a task-agnostic robustness update Δ_{rob}, which reduces adversarial representation drift while explicitly minimizing changes to clean representations relative to the pretrained backbone. **Stage 2:** the learned update is first installed in each descendant by additive composition with fixed $\alpha = 1$. The patched encoder is then frozen. We realign only the task readout using clean unlabeled downstream audio, initializing W from W_0 and keeping $b = b_0$ fixed. The original unpatched descendant provides pseudo-targets, which are matched while encouraging larger geometric margins.

We set $S = 37$ dB, so every feasible perturbation has SNR of at least 37 dB. Headline evaluation uses the worst outcome per example across the fixed seven-attack pool in Table 1. Every attack targets the exact model being evaluated. For our full method, this includes the installed robustness update and realigned readout. The psychoacoustic attack additionally obeys a time–frequency masking constraint. Appendix E.2 characterizes the 37 dB operating point and the psychoacoustic attack.

3.2 Stage 1: Learn once per backbone

Starting from $\Delta = 0$, let $H_\Delta(x)$ denote $1/\sqrt{T}$ times the flattened sequence of T final-layer frame representations of the patched backbone $\theta_b + \Delta$, with H_0 for the original representation. We learn

$$\Delta_{\mathrm{rob}} = \arg\min_{\Delta} \frac{1}{C^2}\, \mathbb{E}_{x\sim\mathcal{D}_s}\left[\max_{\delta\in\Delta_S(x)} \left\|H_\Delta(x+\delta) - \mathrm{sg}[H_\Delta(x)]\right\|_2^2 + \lambda\left\|H_\Delta(x) - H_0(x)\right\|_2^2\right]. \quad (2)$$

We approximate the inner maximization with a PGD attack. The first term minimizes worst-case adversarial drift of the frame representations from their current clean values, with sg blocking gradients through the clean reference. The second term minimizes changes to clean frame representations relative to the original backbone. Unlike using the original clean representation as a single fixed target, our objective separates adversarial stability from clean-representation preservation. Tables 2 and 3 compare against the FARE-style objective and ablate the anchor and stop-gradient choices. We denote the resulting robust backbone by $\theta_r = \theta_b + \Delta_{\mathrm{rob}}$.

3.3 Stage 2: Install per descendant

Stage 2 first installs the learned update in descendant k by additive composition, $\theta_{k,\alpha} = \theta_k + \alpha\Delta_{\mathrm{rob}} = \theta_b + \tau_k + \alpha\Delta_{\mathrm{rob}}$. This retains τ_k in the parameter decomposition, whereas replacing θ_k with θ_r would discard it. All headline results use fixed, untuned $\alpha = 1$; Figure 4b reports a representative sweep over α. Retaining τ_k, however, does not guarantee that Δ_{rob} retains its robustness effect after specialization; we empirically test this in Section 7. Because installation may shift clean representations relative to the original readout, we then freeze $\theta_{k,\alpha}$ and realign the readout by updating only W, keeping $b = b_0$. For each clean calibration utterance $x \sim \mathcal{D}_k^{\mathrm{cal}}$, the original descendant provides the pseudo-target. For classification, $\hat{y}(x) = \arg\max_i [W_0 h_{\theta_k}(x) + b_0]_i$. For ASR, we obtain a pseudo-transcript by collapsing the original descendant's greedy frame-level CTC predictions. For class c and representation z, define the signed geometric margin

$$m_c(z; W, b) = \min_{j\neq c} \frac{\langle w_c - w_j, z\rangle + (b_c - b_j)}{\|w_c - w_j\|_2}. \quad (3)$$

Starting from W_0, realignment minimizes

$$\min_{W} \mathbb{E}_{x\sim\mathcal{D}_k^{\mathrm{cal}}}\left[\mathcal{L}_{\mathrm{task}}(Wh_{\theta_{k,\alpha}}(x)+b_0,\hat{y}(x))+\beta\,\max\big(0,\ \gamma-m_{\hat{y}(x)}(h_{\theta_{k,\alpha}}(x);W,b_0)\big)\right]. \quad (4)$$

For classification, $\mathcal{L}_{\mathrm{task}}$ is cross-entropy. The first term preserves the original descendant's clean predictions on the patched representations, while the second penalizes pseudo-label margins below γ. Each normalized logit gap is the signed distance to a pairwise decision hyperplane, and m_c takes the minimum over competing classes. Neither β nor γ is selected using downstream adversarial examples.

For CTC-based ASR, the prediction-matching term is the CTC loss between the realigned model's output and this pseudo-transcript, while the margin term is applied only to frames assigned a non-blank token by the original descendant. Appendix C.2 gives the exact ASR procedure. Given the learned Δ_{rob}, Stage 2 therefore requires only the descendant weights and clean unlabeled in-distribution calibration audio, with no downstream labels, downstream adversarial examples, or access to the original specialization pipeline. Appendix H further studies sensitivity to the calibration-set size across classification tasks.

4 Analysis: from robust representations to robust predictions

We use two complementary measurements to connect representation stability to robust *predictions*. First, we test whether installing Δ_{rob} reduces adversarial representation drift after specialization, since an update optimized at θ_b need not have the same effect at θ_k. Second, we compare the residual drift with the decision margin, since lower drift alone does not ensure prediction stability. Both measurements use the descendant test split, with attacks targeting the exact model being evaluated.

4.1 Drift reduction after installation

For classification, we measure representation drift using the same seven-attack pool as RA. For each example x, let $\delta_{a,\theta}(x)$ be the valid candidate returned by attack $a\in\mathcal{A}$. We select the candidate with the lowest ground-truth geometric margin,

$$a^\star_\theta(x)\in\arg\min_{a\in\mathcal{A}} m_{y_{\mathrm{gt}}}(h_\theta(x+\delta_{a,\theta}(x));W,b),\qquad \delta^\star_\theta(x)=\delta_{a^\star_\theta(x),\theta}(x). \quad (5)$$

Pooled RA counts an example as robust only if all attacks in the pool fail. To make the drift measurement reflect the same per-example criterion, we retain the returned perturbation with the lowest ground-truth geometric margin. We then define the mean representation drift under this selected perturbation,

$$R_\theta=\mathbb{E}_{x\sim\mathcal{D}_k^{\mathrm{test}}}\left[\|h_\theta(x+\delta^\star_\theta(x))-h_\theta(x)\|_2\right]. \quad (6)$$

The selected perturbation also depends on the evaluated readout (W,b), which we suppress in the notation for brevity. Lower R_θ indicates less attack-induced representation change under the evaluated attack pool.

Limits of drift transfer. Whether additive installation reduces mean drift, $R_{\theta_k+\Delta_{\mathrm{rob}}}<R_{\theta_k}$, is an empirical question, not a consequence of parameter addition. Because the map from parameters to input sensitivity is nonlinear, drift suppression learned at θ_b need not persist after specialization to $\theta_k=\theta_b+\tau_k$. Individual examples may improve even when mean drift does not, while readout realignment can recover margin loss and provide additional robustness without changing the encoder. Drift transfer may depend on specialization, source-to-downstream match, and task difficulty. Table 13 characterizes these factors on the primary backbone; with only four descendants, we treat the comparison as descriptive and do not fit a predictive model. Appendix D also evaluates the same installation procedure on LoRA-specialized descendants.

4.2 A sufficient margin condition for robust predictions

Mean drift alone does not determine robust prediction; its effect depends on the available decision margin. We therefore compare each example's drift with its clean geometric margin. For an example x, define the clean geometric margin $m(x)=m_{y_{\mathrm{gt}}}(h_\theta(x);W,b)$ using Eq. 3. Let $r^\star_\theta(x)=h_\theta(x+\delta^\star_\theta(x))-h_\theta(x)$ denote the representation change under the selected perturbation.

Table 1: **Fixed seven-attack untargeted evaluation at 37 dB.** Accuracy (%, ↑) on the four classification descendants of the primary backbone, WavLM-large. Each attack is generated against the model in its column; pooled RA requires all seven attacks to fail. † marks mechanism-aware adaptive attacks; the psychoacoustic attack additionally satisfies a time–frequency masking constraint. Attack protocol and settings are given in Appendix E.1.

	KWS			IC			SID-250			ER		
Attack	θ_k	Ours	AT ref.	θ_k	Ours	AT ref.	θ_k	Ours	AT ref.	θ_k	Ours	AT ref.
AutoPGD (Croce & Hein, 2020)	19.1	85.8	90.2	18.3	88.1	93.8	3.6	44.3	56.9	5.3	40.5	49.2
C&W (Carlini & Wagner, 2018)	18.9	85.6	90.9	18.1	88.1	94.0	3.6	42.9	57.8	5.5	40.4	48.8
Psychoacoustic (Qin et al., 2019)	23.7	90.3	92.5	21.5	91.7	94.5	7.2	65.1	68.4	7.0	48.9	54.1
Representation drift†	22.3	84.3	92.6	20.4	88.5	93.9	5.8	46.2	65.0	11.8	44.6	56.4
Margin minimization†	19.7	84.9	91.7	19.3	89.5	94.7	4.2	48.8	61.6	6.5	41.5	52.9
Drift + margin†	20.2	83.2	91.8	19.2	87.8	94.3	4.7	45.3	65.2	9.2	37.4	53.7
SPSA (Uesato et al., 2018)	49.4	90.4	92.0	43.6	91.9	95.1	27.5	63.7	68.4	38.1	57.3	58.5
Pooled robust accuracy	**18.3**	**82.1**	**90.1**	**17.5**	**87.6**	**93.8**	**3.2**	**40.9**	**56.8**	**5.3**	**37.1**	**48.6**

Margin criterion. *If x is clean-correct and $m(x) > \|r^\star_\theta(x)\|_2$, every returned perturbation in the seven-attack pool preserves the correct prediction.*

Proof sketch. Let $c = y_{\mathrm{gt}}$ and $r = r^\star_\theta(x)$. For any $j \neq c$, Cauchy–Schwarz gives

$$m_{c,j}(h_\theta(x) + r) \geq m_{c,j}(h_\theta(x)) - \|r\|_2 \geq m(x) - \|r\|_2 > 0.$$

Thus the selected perturbation preserves class c. Since Eq. 5 selects the returned candidate with the lowest ground-truth geometric margin, every other returned candidate also has positive margin and is therefore correct.

This attack-relative condition applies only to the returned perturbations; it does not certify robustness over the full threat set. It links the two mechanisms: Stage 1 targets lower representation drift, while readout realignment encourages larger pseudo-label margins. On the test split, we evaluate true-label margins and report the stable fraction $S_{\mathcal{A}} = \Pr[\, x \text{ clean-correct and } m(x) > \|r^\star_\theta(x)\|_2 \,]$. Because $S_{\mathcal{A}}$ is measured over all test examples, it is directly comparable with pooled RA. Since the condition is sufficient but not necessary, $S_{\mathcal{A}} \leq \mathrm{RA}$; the gap consists of examples that survive all seven attacks without satisfying it. Figure 3 reports the mean clean geometric margin, mean selected representation drift, and $S_{\mathcal{A}}$ for the original descendant, additive installation, and the full method. For CTC-based ASR, the margin argument applies only at frame level, so we evaluate sequence robustness directly using adversarial WER. Appendix G provides additional details on the drift and margin analysis.

5 Experimental setup

We evaluate three pretrained speech backbones spanning contrastive learning, masked prediction with denoising, and cross-modal pretraining. We specialize each backbone to five downstream tasks: keyword spotting (KWS), intent classification (IC), speaker identification (SID-250; a fixed 250-speaker VoxCeleb1 subset), automatic speech recognition (ASR), and emotion recognition (ER). WavLM-large is the primary backbone; wav2vec2-large and SpeechT5 assess whether the method generalizes across backbones. Unless stated otherwise, headline results use descendants obtained by full encoder fine-tuning. LoRA descendants provide an additional specialization regime (Appendix D). Appendix C.1 gives dataset, model, optimization, split, and seed details.

Access-matched comparisons. Post-hoc comparison baselines receive the same deployment-time information as our method: descendant weights and clean unlabeled downstream audio. Encoder-changing baselines receive the same label-free readout realignment to control for readout misalignment; checkpoint interpolation and TIES select merge settings using only the clean validation split. Robust-first adaptation and task-specific AT use downstream supervision and therefore serve as references rather than access-matched alternatives.

Robustness evaluation. All adversarial examples target the exact model being evaluated. Classification uses a fixed seven-attack pool: AutoPGD, C&W, an SNR-constrained psychoacoustic attack, three mechanism-aware adaptive attacks, and query-limited SPSA. An example is robust only if all

Table 2: **Main comparison: post-hoc robustness after specialization.** Classification cells report clean / pooled robust accuracy (%, ↑); ASR reports clean / adversarial WER (%, ↓). ± terms are bootstrap standard errors from 1000 shared test-set resamples. T is the mean percentage of task-specific AT gain recovered across the four classification tasks; ASR is reported separately in WER and excluded from T. Encoder-changing comparison baselines receive the same label-free realignment; ‡ marks merge coefficients selected using clean validation only. WavLM-large is the primary backbone; wav2vec2-large and SpeechT5 assess generalization across backbones. Access keys: U clean unlabeled downstream audio, L labeled downstream audio, P original specialization pipeline, AT downstream adversarial training.

		Acc. (%) ↑					WER (%) ↓
Method	Access	KWS	IC	SID-250	ER	$\overline{T}$ (%) ↑	ASR
WavLM-large *(primary backbone)*							
θ_k (standard fine-tuning)	–	97.8$_{\pm 0.3}$ / 18.3$_{\pm 0.7}$	99.1$_{\pm 0.2}$ / 17.5$_{\pm 0.6}$	91.4$_{\pm 0.6}$ / 3.2$_{\pm 0.4}$	68.6$_{\pm 0.6}$ / 5.3$_{\pm 0.3}$	0.0	8.2$_{\pm 0.3}$ / 91.5$_{\pm 1.0}$
Post-hoc routes (no downstream labels)							
Encoder replacement ($\theta_k \to \theta_r$) + realignment	U	76.5$_{\pm 0.8}$ / 60.8$_{\pm 1.0}$	81.4$_{\pm 0.6}$ / 66.5$_{\pm 0.7}$	65.5$_{\pm 1.1}$ / 21.2$_{\pm 0.9}$	48.6$_{\pm 0.7}$ / 19.1$_{\pm 0.5}$	47.2	19.6$_{\pm 0.4}$ / 36.7$_{\pm 0.6}$
Checkpoint interpolation (Ilharco et al., 2022)‡ + realignment	U	91.5$_{\pm 0.3}$ / 66.9$_{\pm 0.9}$	92.0$_{\pm 0.4}$ / 73.8$_{\pm 0.7}$	78.5$_{\pm 1.0}$ / 31.5$_{\pm 1.1}$	58.9$_{\pm 0.6}$ / 28.0$_{\pm 0.5}$	61.7	15.4$_{\pm 0.4}$ / 48.1$_{\pm 0.7}$
TIES merge (Yadav et al., 2023)‡ + realignment	U	93.8$_{\pm 0.4}$ / 76.2$_{\pm 0.7}$	95.1$_{\pm 0.4}$ / 84.1$_{\pm 0.6}$	81.2$_{\pm 0.8}$ / 34.2$_{\pm 1.0}$	62.1$_{\pm 0.7}$ / 32.5$_{\pm 0.6}$	72.1	12.3$_{\pm 0.2}$ / 35.8$_{\pm 0.5}$
FARE-style update (Schlarmann et al., 2024) + realignment	U	90.6$_{\pm 0.6}$ / 76.9$_{\pm 0.8}$	82.9$_{\pm 0.6}$ / 71.0$_{\pm 0.7}$	76.0$_{\pm 0.9}$ / 28.6$_{\pm 1.1}$	55.2$_{\pm 0.7}$ / 26.9$_{\pm 0.6}$	62.3	20.5$_{\pm 0.5}$ / 38.4$_{\pm 0.6}$
Patch only ($\theta_k + \Delta_{\text{rob}}$)	–	94.0$_{\pm 0.3}$ / 78.5$_{\pm 0.7}$	91.3$_{\pm 0.5}$ / 80.4$_{\pm 0.8}$	73.3$_{\pm 1.0}$ / 31.4$_{\pm 1.0}$	46.3$_{\pm 0.8}$ / 23.7$_{\pm 0.6}$	65.3	14.1$_{\pm 0.4}$ / 37.2$_{\pm 0.6}$
Ours (install + realign)	**U**	95.2$_{\pm 0.4}$ / 82.1$_{\pm 0.6}$	98.2$_{\pm 0.2}$ / 87.6$_{\pm 0.5}$	84.1$_{\pm 0.8}$ / 40.9$_{\pm 1.1}$	63.4$_{\pm 0.6}$ / 37.1$_{\pm 0.7}$	**81.1**	11.0$_{\pm 0.3}$ / 30.7$_{\pm 0.5}$
Robust-first / supervised references							
Robust-first (FARE-style): frozen encoder + supervised head	L	80.0$_{\pm 0.7}$ / 62.4$_{\pm 1.0}$	82.2$_{\pm 0.6}$ / 64.6$_{\pm 0.7}$	71.5$_{\pm 1.1}$ / 19.2$_{\pm 0.8}$	53.8$_{\pm 0.7}$ / 16.7$_{\pm 0.5}$	44.8	18.3$_{\pm 0.4}$ / 35.8$_{\pm 0.7}$
Robust-first (FARE-style): full clean fine-tuning	L+P	96.6$_{\pm 0.3}$ / 34.2$_{\pm 0.9}$	97.4$_{\pm 0.3}$ / 29.6$_{\pm 0.8}$	88.2$_{\pm 0.8}$ / 11.7$_{\pm 0.8}$	64.1$_{\pm 0.6}$ / 9.9$_{\pm 0.3}$	16.1	9.8$_{\pm 0.3}$ / 67.1$_{\pm 0.8}$
Task-specific waveform AT (Madry et al., 2018) *(reference)*	L+AT	94.5$_{\pm 0.4}$ / 90.1$_{\pm 0.5}$	96.6$_{\pm 0.3}$ / 93.8$_{\pm 0.4}$	79.3$_{\pm 0.9}$ / 56.8$_{\pm 1.2}$	60.2$_{\pm 0.7}$ / 48.6$_{\pm 0.8}$	100.0	14.3$_{\pm 0.4}$ / 21.8$_{\pm 0.5}$
wav2vec2-large							
θ_k (standard fine-tuning)	–	96.3$_{\pm 0.3}$ / 14.7$_{\pm 0.7}$	98.2$_{\pm 0.2}$ / 18.9$_{\pm 0.6}$	86.3$_{\pm 0.7}$ / 2.8$_{\pm 0.4}$	65.0$_{\pm 0.6}$ / 0.2$_{\pm 0.1}$	0.0	9.6$_{\pm 0.3}$ / 93.9$_{\pm 1.1}$
FARE-style update + realignment	U	87.1$_{\pm 0.6}$ / 66.0$_{\pm 0.9}$	90.3$_{\pm 0.5}$ / 70.6$_{\pm 0.8}$	71.8$_{\pm 1.1}$ / 31.5$_{\pm 1.1}$	54.7$_{\pm 0.8}$ / 24.5$_{\pm 0.6}$	62.9	23.7$_{\pm 0.5}$ / 44.5$_{\pm 0.7}$
Ours (install + realign)	**U**	95.4$_{\pm 0.4}$ / 72.2$_{\pm 0.8}$	97.1$_{\pm 0.3}$ / 78.4$_{\pm 0.6}$	81.6$_{\pm 0.8}$ / 42.1$_{\pm 1.0}$	62.7$_{\pm 0.7}$ / 34.8$_{\pm 0.6}$	**78.4**	12.2$_{\pm 0.3}$ / 29.7$_{\pm 0.4}$
Task-specific waveform AT *(reference)*	L+AT	92.8$_{\pm 0.3}$ / 88.9$_{\pm 0.7}$	92.9$_{\pm 0.4}$ / 86.5$_{\pm 0.5}$	73.8$_{\pm 1.0}$ / 59.7$_{\pm 1.1}$	56.8$_{\pm 0.7}$ / 43.9$_{\pm 0.7}$	100.0	14.5$_{\pm 0.4}$ / 20.8$_{\pm 0.4}$
SpeechT5							
θ_k (standard fine-tuning)	–	97.0$_{\pm 0.3}$ / 11.5$_{\pm 0.6}$	96.1$_{\pm 0.3}$ / 13.9$_{\pm 0.6}$	83.2$_{\pm 0.9}$ / 1.0$_{\pm 0.2}$	65.4$_{\pm 0.6}$ / 1.2$_{\pm 0.1}$	0.0	13.0$_{\pm 0.4}$ / 97.2$_{\pm 1.1}$
FARE-style update + realignment	U	85.5$_{\pm 0.6}$ / 66.9$_{\pm 0.9}$	81.4$_{\pm 0.6}$ / 67.0$_{\pm 0.8}$	70.5$_{\pm 1.0}$ / 22.1$_{\pm 0.9}$	52.8$_{\pm 0.7}$ / 18.9$_{\pm 0.5}$	58.9	27.4$_{\pm 0.5}$ / 52.6$_{\pm 0.7}$
Ours (install + realign)	**U**	94.9$_{\pm 0.4}$ / 74.2$_{\pm 0.8}$	93.6$_{\pm 0.5}$ / 80.9$_{\pm 0.7}$	78.8$_{\pm 0.9}$ / 34.8$_{\pm 1.0}$	63.7$_{\pm 0.6}$ / 28.0$_{\pm 0.6}$	**78.8**	15.2$_{\pm 0.4}$ / 35.0$_{\pm 0.6}$
Task-specific waveform AT *(reference)*	L+AT	92.4$_{\pm 0.4}$ / 89.6$_{\pm 0.6}$	91.8$_{\pm 0.5}$ / 88.9$_{\pm 0.5}$	74.5$_{\pm 1.0}$ / 50.1$_{\pm 1.2}$	58.9$_{\pm 0.7}$ / 36.1$_{\pm 0.6}$	100.0	15.1$_{\pm 0.4}$ / 28.4$_{\pm 0.5}$

seven attacks fail. ASR uses the same pool with ASR-specific objectives; Table 2 reports adversarial WER. Every retained perturbation is verified against Eq. 1. The adaptive attacks, SPSA, and the attack-strength sweep in Fig. 4a provide complementary checks against gradient obfuscation. Appendix E gives the full robustness-evaluation protocol, perceptual and psychoacoustic details, and a targeted-ASR scope check.

Metrics and uncertainty. Classification reports clean accuracy (CA) and pooled robust accuracy (RA); ASR reports clean and adversarial WER. We also report T, the mean percentage of task-specific AT gain recovered across the four classification tasks. Table 2 reports bootstrap standard errors from 1000 shared test-set resamples, and Appendix C.1 reports variability across three independent specialization seeds. The calibration split is reserved before downstream fine-tuning, disjoint from the validation and test sets, and used only for readout realignment and post-hoc analyses. Validation is reserved for explicitly reported reference selections, including merge coefficients.

6 Robustness transfer after specialization

At fixed, untuned $\alpha = 1$, post-hoc installation improves seven-attack RA by 48.7 points on average across the 12 classification descendants (Table 2), with a worst-task clean-accuracy loss of 7.3 points. The method recovers 79.5% of the task-specific AT gain. On the primary backbone, additive installation and readout realignment are complementary: additive installation provides most of the robustness gain but reduces clean accuracy by 13.0 points on average, while realignment recovers 9.0 of those points (69.2%) and adds 8.4 RA points on average with the encoder frozen. Without the robustness update, realignment provides only 4.0% of the full robustness gain (Table 3), indicating that readout repair alone explains little of the improvement.

What the post-hoc baselines test. Encoder replacement tests whether a robust backbone can substitute for the descendant; it underperforms additive installation despite the same label-free realignment, consistent with discarding τ_k. Checkpoint interpolation and TIES test generic parameter-space composition. Using the same additive installation and readout realignment, the FARE-style baseline replaces our Stage 1 learning objective with a FARE-style objective. It recovers 62.3% of task-specific AT gain on the primary backbone. Together with patch-only and realignment-only controls,

these comparisons distinguish *what* update is learned, *how* it is installed, and *whether* the readout is realigned.

Order of operations and amortization. Robust-first baselines introduce robustness before specialization. On the primary backbone, freezing the robust encoder yields higher robust accuracy but lower clean task performance than subsequent clean fine-tuning, which improves task fit but substantially reduces robustness. Our method reverses this order: Stage 1 is learned once per backbone and Stage 2 is applied to each already-specialized descendant. Because the Stage 1 cost is amortized, mean costs across backbones and tasks break even with task-specific AT at $K \approx 2.2$ descendants and fall to 16% of task-specific AT cost per descendant at $K = 15$ (Appendix F).

7 From restored stability to robust predictions

Following the analysis in Section 4, Fig. 3 examines drift and margins on the primary backbone, WavLM-large; Table 2 reports robustness results for wav2vec2-large and SpeechT5. Drift decreases on all four descendants. The full method reduces mean drift by factors of 12.8, 9.4, 4.5, and 4.8 on KWS, IC, SID-250, and ER, respectively. The drift measurements use the selected perturbations defined in Section 4.1. Almost all drift reduction occurs during additive installation; readout realignment does not modify the encoder. Realignment addresses margin loss. Installation reduces mean clean geometric margin by 0.1–0.4, and realignment more than recovers this loss on three of the four tasks. Additive installation and readout realignment thus address opposite sides of the sufficient margin criterion. The stable fraction $S_{\mathcal{A}}$ increases after installation and again after realignment on every task. For the full method, $S_{\mathcal{A}}$ is within 2.5 RA points of pooled RA on KWS, IC, and ER, while the bound is looser on SID-250 ($S_{\mathcal{A}} = 33.1\%$ versus $\mathrm{RA} = 40.9\%$).

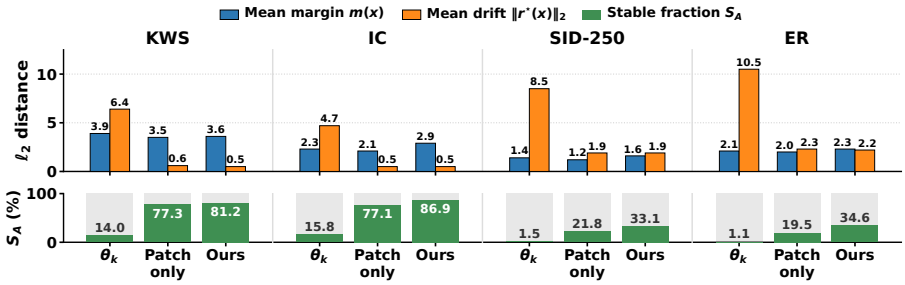

Figure 3: **How restored stability becomes a robust prediction.** *Upper:* mean clean geometric margin $m(x)$ and mean representation drift R_θ under the selected perturbation, for θ_k, patch only, and the full method. *Lower:* stable fraction $S_{\mathcal{A}}$, the fraction of test examples that are clean-correct and satisfy the margin criterion, hence a lower bound on pooled robust accuracy. The upper panel reports mean margin and mean drift across the test split, whereas $S_{\mathcal{A}}$ compares them per utterance; thus, mean margin can exceed mean drift even when many individual examples fail the criterion.

Audits: attack budget and installation strength. On SID-250, RA decreases monotonically as attack strength increases for every configuration (Fig. 4a), providing an additional check against gradient masking. The installation sweep (Fig. 4b) supports fixed $\alpha = 1$: increasing α to 1.25 improves RA by only 0.4 points while reducing clean accuracy by 4.0 points. Beyond $\alpha = 1.25$, both CA and RA fall sharply.

8 Ablations and controls

Table 3 reports per-task ablations on the primary backbone. In Stage 1, replacing adversarial perturbations with random SNR-matched noise (Δ_{noise}) lowers mean RA by 47.6 points. Removing stop-gradient costs 6.7 mean RA points, while removing the clean anchor lowers mean CA by 20.6. In Stage 2, removing the geometric-margin term lowers mean RA by 5.4 points; using an unnormalized logit margin (Eq. 3 without the denominator) lowers it by 5.5. Both margin ablations hurt SID-250 and ER most, the tasks with the smallest post-installation mean margin–drift gaps (Fig. 3).

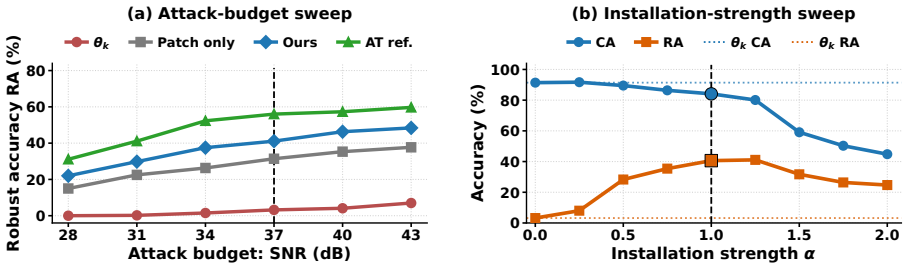

Figure 4: **Attack-budget and installation-strength audits.** Both panels use the SID-250 descendant of WavLM-large, where installation incurs the largest clean-accuracy loss (7.3 points), placing the method under its least favourable observed trade-off. *(a)* Robust accuracy across attack budgets in 3 dB steps; the dashed line marks the 37 dB operating point. *(b)* Clean and robust accuracy across α; the dashed line marks the fixed, untuned operating point $\alpha = 1$.

Table 3: **Ablations, controls, and supervision references on the primary backbone.** Classification cells report clean / pooled robust accuracy (%, ↑); ASR reports clean / adversarial WER (%, ↓). Stage 1, Stage 2, and control rows use no downstream labels; supervision-reference rows replace pseudo-targets with the indicated label fraction under the same W-only fit, with $b = b_0$ fixed.

	Acc. (%) ↑				WER (%) ↓
Configuration	KWS	IC	SID-250	ER	ASR
Ours (full method)	95.2 / 82.1	98.2 / 87.6	84.1 / 40.9	63.4 / 37.1	11.0 / 30.7
Stage 1 robustness update					
w/o stop-gradient	91.2 / 72.8	93.5 / 78.3	77.0 / 36.2	56.3 / 33.5	17.3 / 44.1
w/o clean anchor (λ=0)	74.8 / 53.6	78.1 / 61.1	60.0 / 28.1	45.8 / 27.5	26.2 / 60.5
w/o adversary (Δ_{noise})	96.1 / 23.7	98.1 / 21.3	80.7 / 4.1	64.0 / 8.1	9.4 / 81.5
Stage 2 readout realignment					
w/o realignment (patch only)	94.0 / 78.5	91.3 / 80.4	73.3 / 31.4	46.3 / 23.7	14.1 / 37.2
w/o margin (β=0)	95.8 / 77.9	98.1 / 83.5	84.3 / 33.7	63.5 / 31.2	10.8 / 33.6
Unnormalized logit margin	95.2 / 77.8	97.4 / 83.1	83.9 / 33.4	63.1 / 31.3	12.0 / 34.2
Supervision references					
Ours (5% supervised realignment)	95.2 / 82.0	98.3 / 87.2	84.2 / 40.7	63.3 / 37.4	11.0 / 30.6
Ours (10% supervised realignment)	95.5 / 82.5	98.2 / 88.1	84.3 / 40.0	63.5 / 38.2	11.1 / 30.4
Ours (100% supervised realignment)	96.9 / 84.9	98.9 / 88.4	85.4 / 42.6	64.8 / 41.5	10.7 / 28.7
Control					
Realignment only (unpatched θ_k)	96.2 / 21.6	98.4 / 19.4	90.0 / 5.1	67.9 / 6.3	8.4 / 87.2

Label-free realignment is within 1.1 RA points of the 10%-label reference on every classification task. Realignment alone on the unpatched descendant recovers only 4.0% of the full method's mean RA gain.

9 Limitations

Our evaluation focuses on digital waveform perturbations under the stated SNR-constrained threat model; physical playback and re-recording are considered a separate over-the-air setting. Table 10 characterizes how detectable the perturbations are using objective metrics and ABX discrimination. The margin criterion is attack-relative and sufficient for the returned perturbations in the evaluated pool. For CTC-based ASR, it applies at the frame-level projection, so sequence robustness is evaluated directly using adversarial WER. Finally, Δ_{rob} is learned per backbone, allowing its Stage 1 cost to be amortized across all descendants that share that backbone.

10 Conclusion

We study adversarial robustness after specialization: robustness is added post hoc to descendants. We learn one task-agnostic update per backbone from unlabeled source audio, install it at a fixed, untuned scale, and realign the readout label-free. Across 12 classification descendants, seven-attack RA improves by 48.7 points on average, with a 7.3-point worst-task CA loss and 79.5% recovery of the task-specific AT gain. Installation requires no downstream labels, original fine-tuning pipeline, per-descendant AT, or additional inference-time computation. Our analysis links reduced adversarial representation drift to robust predictions through an attack-relative sufficient margin condition. These results show that robustness can be learned per backbone and installed after specialization.

Reproducibility statement. We will release the per-backbone Δ_{rob} updates, configurations, fixed SID-250 speaker list, attack implementations, and robustness-evaluation code. Appendices C–H provide the experimental setup, specialization controls, robustness-evaluation protocol, compute accounting, diagnostic procedures, and calibration-data sensitivity.

Ethics statement. The perceptual evaluation in Appendix E.2 was a listening study involving 12 participants, each completing 50 ABX trials. Participants were volunteers and provided informed consent to participate. The study was limited to perceptual discrimination of the audio examples described in Appendix E.2, and no identifying participant information was collected, reported, or released. The need for ethics review was assessed under the applicable institutional requirements, and formal review was not required. The remaining experiments use established speech datasets and involve no additional participant data collection.

AI Use Statement. Generative AI tools were used as assistants during manuscript preparation, primarily for language refinement, organization, LaTeX and table formatting, reference verification, numerical consistency checks, and feedback on clarity and presentation. The research problem, methodology, experimental design, implementation, experiments, analyses, and scientific conclusions were developed and carried out by the authors. The authors take full responsibility for the content of the paper.

REFERENCES

Junyi Ao, Rui Wang, Long Zhou, Chengyi Wang, Shuo Ren, Yu Wu, Shujie Liu, Tom Ko, Qing Li, Yu Zhang, Zhihua Wei, Yao Qian, Jinyu Li, and Furu Wei. SpeechT5: Unified-modal encoder-decoder pre-training for spoken language processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. Common Voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference (LREC)*, pp. 4218–4222, 2020. URL https://aclanthology.org/2020.lrec-1.520/.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan. IEMOCAP: Interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42(4):335–359, 2008. doi: 10.1007/s10579-008-9076-6.

Ruisi Cai, Zhenyu Zhang, and Zhangyang Wang. Robust weight signatures: Gaining robustness as easy as patching weights? In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202, pp. 3495–3506, 2023.

Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *IEEE Security and Privacy Workshops (SPW)*, 2018.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022. doi: 10.1109/JSTSP.2022.3188113.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Junhao Dong, Xinghua Qu, Cong Zhang, Qi Rong Sua, Nguyen Duc Thai, Wenbo Pan, Xinfeng Li, Tongliang Liu, Piotr Koniusz, and Yew-Soon Ong. Tug-of-war no more: Harmonizing accuracy and robustness in vision-language models via stability-aware task vector merging. In *International Conference on Learning Representations (ICLR)*, 2026.

Md Zarif Hossain and Ahmed Imteaj. Sim-CLIP: Unsupervised siamese adversarial fine-tuning for robust and semantically-rich vision-language models. In *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2026. URL https://arxiv.org/abs/2407.14971.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2023.

Sonal Joshi, Saurabh Kataria, Yiwen Shao, Piotr Żelasko, Jesús Villalba, Sanjeev Khudanpur, and Najim Dehak. Defense against adversarial attacks on hybrid speech recognition system using adversarial fine-tuning with denoiser. In *Interspeech*, pp. 5035–5039, 2022. doi: 10.21437/Interspeech.2022-10977.

Borna Khodabandeh, Amirabbas Afzali, Amirhossein Afsharrad, Seyed Shahabeddin Mousavi, Sanjay Lall, Sajjad Amini, and Seyed-Mohsen Moosavi-Dezfooli. LORE: Lagrangian-optimized robust embeddings for visual encoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025. doi: 10.52202/085713-2752.

Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. Speech model pre-training for end-to-end spoken language understanding. In *Interspeech 2019*, pp. 814–818, 2019. doi: 10.21437/Interspeech.2019-2396.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

Chengzhi Mao, Scott Geng, Junfeng Yang, Xin Wang, and Carl Vondrick. Understanding zero-shot adversarial robustness for large-scale models. In *International Conference on Learning Representations (ICLR)*, 2023.

Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. VoxCeleb: A large-scale speaker identification dataset. In *Interspeech 2017*, pp. 2616–2620, 2017. doi: 10.21437/Interspeech.2017-950.

Raphael Olivier and Bhiksha Raj. Sequential randomized smoothing for adversarially robust speech recognition. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6372–6386, 2021.

Raphael Olivier, Hadi Abdullah, and Bhiksha Raj. Watch what you pretrain for: Targeted, transferable adversarial examples on self-supervised speech recognition models. *arXiv preprint arXiv:2209.13523*, 2022.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. LibriSpeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015. doi: 10.1109/ICASSP.2015.7178964.

Yao Qin, Nicholas Carlini, Garrison Cottrell, Ian Goodfellow, and Colin Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

Christian Schlarmann, Naman Deep Singh, Francesco Croce, and Matthias Hein. Robust CLIP: Unsupervised adversarial fine-tuning of vision embeddings for robust large vision-language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pp. 43685–43704, 2024.

Lea Schönherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding. In *Network and Distributed System Security Symposium (NDSS)*, 2019.

Ali Shafahi, Parsa Saadatpanah, Chen Zhu, Amin Ghiasi, Christoph Studer, David Jacobs, and Tom Goldstein. Adversarially robust transfer learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Jonathan Uesato, Brendan O'Donoghue, Pushmeet Kohli, and Aaron van den Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

Pete Warden. Speech Commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018. URL `https://arxiv.org/abs/1804.03209`.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

Haibin Wu, Bo Zheng, Xu Li, Xixin Wu, Hung-yi Lee, and Helen Meng. Characterizing the adversarial vulnerability of speech self-supervised learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

Shutong Wu, Jiongxiao Wang, Wei Ping, Weili Nie, and Chaowei Xiao. Defending against adversarial audio via diffusion model. In *International Conference on Learning Representations (ICLR)*, 2023.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-Merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu-Hsien Huang, Wei-Cheng Tseng, Ko-tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung-yi Lee. SUPERB: Speech processing universal PERformance benchmark. In *Interspeech 2021*, pp. 1194–1198, 2021. doi: 10.21437/Interspeech.2021-1775.

A Extended related work

Robust encoders and robust transfer. TeCoA (Mao et al., 2023), FARE/Robust CLIP (Schlarmann et al., 2024), and LORE (Khodabandeh et al., 2025) make generic visual encoders robust for downstream use. In downstream systems that freeze the visual encoder, this robustness can be deployed by replacing the original encoder. Our setting differs because the encoder itself has already been specialized to the target task. Robust transfer learning shows that frozen robust features can support downstream robustness, whereas unrestricted end-to-end adaptation can erode it (Shafahi et al., 2020). We address the reverse order of operations.

Objective-level comparison. FARE matches an adapted encoder's adversarial embedding to the original encoder's clean embedding without labels. For a candidate update Δ and a fixed perturbation δ, write $a = H_\Delta(x+\delta)$, $b = H_\Delta(x)$, and $c = H_0(x)$. Its penalty satisfies

$$\|a-c\|_2^2 = \|a-b\|_2^2 + \|b-c\|_2^2 + 2\langle a-b, b-c\rangle. \tag{7}$$

In contrast, the per-example expression in Eq. 2, before scaling by C^{-2}, is $\|a - \mathrm{sg}[b]\|_2^2 + \lambda\|b - c\|_2^2$. It uses a moving clean target for adversarial stability and a separately weighted fixed clean anchor. The expansion compares scalar values at a common δ; it does not imply identical training gradients, because stop-gradient removes the clean branch's contribution to the stability update. The objectives can also select different inner maximizers. Table 2 compares a FARE-style update under the same installation and realignment procedure, while Table 3 separately removes stop-gradient and the clean anchor. These comparisons test the objective and its components, not the cross term alone. LORE also controls clean-embedding deviation, using a proximity constraint and learned Lagrange multipliers while retaining the original clean embedding as its adversarial target. SimCLIP (Hossain & Imteaj, 2026) also aligns clean and adversarial views, using a cosine objective with symmetric stop-gradient. Our combination uses one-sided stop-gradient, an explicit fixed clean anchor, and additive installation into specialized encoders; we do not claim either clean anchoring or stop-gradient alone as novel.

Weight-space transfer and robustness patches. Task arithmetic, interpolation, model soups, and interference-aware merging motivate reuse of fine-tuning displacements (Ilharco et al., 2023; 2022; Wortsman et al., 2022; Yadav et al., 2023). Robust Weight Signatures (RWS) (Cai et al., 2023) provide a direct precedent: they extract corruption-specific signatures by projecting weight displacements from a shared initialization, using paired standard and corruption-robust classifiers. These signatures transfer across datasets and can patch models after further adaptation. RWS constructs its signatures from supervised source-task models and targets common corruptions; in contrast, we learn a task-agnostic update from a generic backbone using unlabeled audio and adversarial input perturbations. PISTOLE (Dong et al., 2026) merges natural and adversarial task vectors using labeled calibration examples, input attacks, and adversarial parameter trajectories. Its downstream VLM experiments replace the vision encoder with the merged encoder without retraining the downstream system. This differs from adding a shared update to each target-specialized encoder while retaining its specialization displacement.

Adversarial robustness in pretrained speech. Waveform attacks can induce targeted ASR failures (Carlini & Wagner, 2018; Schönherr et al., 2019; Qin et al., 2019), and attacks on self-supervised speech models can transfer across downstream tasks and independently fine-tuned ASR systems (Wu et al., 2022; Olivier et al., 2022). Defenses include adversarial fine-tuning with a denoiser (Joshi et al., 2022), randomized smoothing (Olivier & Raj, 2021), and diffusion-based purification (Wu et al., 2023). For purification, adaptive evaluation must target the entire defended pipeline, using appropriate backpropagation, gradient approximations, or expectation over transformation (Athalye et al., 2018). AudioPure explicitly evaluates adaptive attacks and leaves the classifier's weights fixed while transforming its input; the resulting internal representations can change. Our method instead modifies the specialized encoder and realigns its readout once, without additional inference-time computation. We do not evaluate combining AudioPure with our method.

B Positioning and access assumptions

Table 4 compares representative adversarial-robustness routes by the requirements for constructing robustness, whether the construction cost is amortized across targets, and whether robustness can be installed after encoder specialization. We call a route *amortized* when one robustness construction serves multiple downstream targets; *post-specialization installation* means a direct weight-space intervention on an already-specialized encoder without target AT or replacement by a generic robust encoder. RWS (Cai et al., 2023) is a related weight-patching precedent rather than a direct adversarial-robustness baseline: it studies common image corruptions, so we discuss it here but omit it from the table.

Our method combines unlabeled source construction, amortization, and post-specialization installation: Stage 1 incurs a one-time cost per backbone and is reused across independently specialized descendants, while Stage 2 installs the learned update in each descendant and realigns only the readout using clean unlabeled calibration audio. It requires no target labels, target adversarial examples, or original specialization pipeline. FARE and PISTOLE also amortize robustness, but their published routes replace a generic or frozen vision encoder rather than patch an encoder that was itself specialized. The frozen robust-extractor reference in Shafahi et al. (2020) uses labeled target fitting, whereas task-specific AT requires separate robust training for each target. The FARE row describes its published encoder-replacement route; the FARE-style baseline in Table 2 instead uses our additive protocol.

Table 4: **Access and order-of-operations comparison for adversarial-robustness routes.** Symbols ✓/✗ indicate whether each property applies to the published route; "–" denotes no separate source stage.

	Robustness construction			Target application		
Approach	Source labels	Robust checkpoint	Amortized across targets	Post-spec. install	Target labels	Target AT
PISTOLE (Dong et al., 2026)	✓	✓	✓	✗	✗	✗
FARE (Schlarmann et al., 2024)	✗	✗	✓	✗	✗	✗
Frozen robust extractor (Shafahi et al., 2020)	✓	✓	✓	✗	✓	✗
Task-specific AT (Madry et al., 2018)	–	–	✗	✗	✓	✓
Ours (install + realign)	✗	✗	✓	✓	✗	✗

C Experimental details

C.1 Backbones, downstream tasks, and optimization

Main descendants use full encoder fine-tuning; Appendix D isolates LoRA as a separate parameter-efficient specialization regime.

Backbones. We evaluate WavLM-large, wav2vec2-large, and SpeechT5, spanning masked-prediction-with-denoising, contrastive, and cross-modal pretraining. All three use convolutional waveform front ends. Table 5 reports pretraining sources and deployed speech-encoder parameter counts; the SpeechT5 decoder is not deployed. The robustness update is learned independently in each backbone's parameter space, so "learned once" means once per backbone rather than one update shared across architectures.

Table 5: **Backbones and pretraining.** Pretraining objectives, data, and deployed speech-encoder parameter counts for the three evaluated backbones.

Backbone	Pretraining objective	Pretraining data	Deployed params
WavLM-large *(primary backbone)*	masked prediction + denoising	60k h Libri-Light + 10k h GigaSpeech + 24k h VoxPopuli	316.6M
wav2vec2-large	contrastive prediction of quantized targets	960 h LibriSpeech	317.4M
SpeechT5	unified speech/text seq2seq + cross-modal VQ	960 h LibriSpeech + LibriSpeech LM	94.4M

Robustification and calibration audio. Stage 1 uses a randomly selected 100-h subset of Common Voice v26 English (Ardila et al., 2020) as $\mathcal{D}_s$, with labels ignored. The normalization constant

C is computed on a separate 2-h subset of the Common Voice v26 English validation split, held out from Stage 1 optimization. For each downstream task, we reserve 45 minutes from the task's training split as $\mathcal{D}_k^{\mathrm{cal}}$ before specialization and use the remaining training data to fine-tune the descendant. The calibration audio is therefore unseen during specialization and disjoint from validation and test. The readout-realignment step of Stage 2 uses only these waveforms and the original descendant's predictions, never its ground-truth labels.

Downstream protocols. Task definitions and splits follow SUPERB (Yang et al., 2021) where applicable. The downstream datasets are Speech Commands (Warden, 2018), Fluent Speech Commands (Lugosch et al., 2019), VoxCeleb1 (Nagrani et al., 2017), IEMOCAP (Busso et al., 2008), and LibriSpeech (Panayotov et al., 2015). Table 6 reports class counts, chance levels, and train/test utterance counts before the calibration holdout; validation counts are omitted for compactness. For speaker identification, we use SID-250, a fixed 250-speaker subset of the VoxCeleb1/SUPERB SID task. The identities are sampled once with a fixed seed and held fixed across all backbones, methods, attacks, ablations, and specialization seeds. SID-250 is a controlled evaluation protocol rather than the full 1,251-speaker SUPERB SID benchmark; the selected speaker IDs will be released with the experimental configuration. IEMOCAP counts are per fold under five-fold leave-one-session-out evaluation. ASR specialization uses the 100-h LibriSpeech labeled subset and standard development and test subsets. No downstream task trains on the Stage 1 robustification source.

Table 6: **Downstream tasks and evaluation splits.** Metrics, class counts/chance levels, train/test utterance counts, and split protocols used for each task.

Task	Dataset	Metric	Classes / chance	Train / test	Protocol
Keyword spotting (KWS)	Speech Commands	Acc. ↑	12 / 8.33%	51.1k / 3.1k	SUPERB fixed train/dev/test
Intent classification (IC)	FSC	Acc. ↑	31 / 3.23%	23.1k / 3.8k	SUPERB fixed train/dev/test
Speaker identification (SID-250)	VoxCeleb1	Acc. ↑	250 / 0.40%	27.9k / 1.8k	fixed-seed 250-speaker subset
Emotion recognition (ER)	IEMOCAP	Acc. ↑	4 / 25.0%	≈4.4k / ≈1.1k per fold	5-fold leave-one-session-out
Speech recognition (ASR)	LibriSpeech	WER ↓	–	28.5k / 2.6k	100-h labeled train subset; standard dev/test

Optimization. Robustification, specialization, and realignment use AdamW with 15% warmup; Table 7 lists their learning rates, epochs, batch sizes, and objective parameters. Stage 1 updates all encoder parameters for four epochs with batch size 10; main specialization updates the trainable encoder layers and task readout (W, b) for four epochs with batch size 16, while any layers frozen by the specialization setup remain frozen. During the readout-realignment step of Stage 2, pseudo-labels from θ_k and representations from the frozen $\theta_{k,\alpha}$ are generated once with batch size 16 and cached; the linear readout is then fit for 20 epochs with batch size 64. The encoder remains frozen during the readout fit, so its cached representations can be reused. Labeled-refit controls use the same head-only optimization with true labels. The FARE-style baseline uses the same source data and optimization settings as Stage 1, changing only the objective. Both robust-first references reuse this FARE-style checkpoint, either freezing the encoder for supervised readout fitting or performing full supervised clean fine-tuning.

Table 7: **Optimization settings.** Hyperparameters for robustness learning, downstream specialization, and post-hoc readout realignment; all optimization phases use AdamW with 15% warmup.

	Robustification ($\theta_b \to \theta_r$)	Specialization (θ_k)	Realignment (ours)
Learning rate	5×10^{-5}	1×10^{-4}	1×10^{-4}
Epochs	4	4	20 (cached head fit)
Batch size	10	16	16 (encoder pass); 64 (head fit)
Attack during training	PGD, $S = 37$ dB, 10 steps	none	none
λ (clean anchor)	0.5	–	–
β (margin weight), γ (target)	–	–	1.0, 0.5 (defaults)
Supervision	none	labels	pseudo-labels $\hat{y}$
Unlabeled audio used	Common Voice, 100 h	–	45 min held-out train audio
Trainable parameters	all encoder parameters	encoder + readout (W, b)	W only ($b = b_0$ fixed)

Task-specific AT reference. Starting from θ_b, each task-specific AT reference minimizes

$$\min_{\theta,W,b} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta_S(x)} \mathcal{L}_{\mathrm{task}}\big(g_{W,b}(h_\theta(x+\delta)), y\big) + 0.5\, \mathcal{L}_{\mathrm{task}}\big(g_{W,b}(h_\theta(x)), y\big) \right], \tag{8}$$

with $S = 37$ dB. The inner maximization uses 10-step PGD; the outer optimization uses AdamW for four epochs with learning rate 5×10^{-5} and batch size 10. Here, $\mathcal{L}_{\text{task}}$ is cross-entropy for classification and CTC loss for ASR.

Specialization-seed variability. Table 8 reports clean/robust accuracy ranges across three independent specialization seeds for each WavLM-large classification descendant. With three seeds we report ranges rather than significance tests; these quantify training-seed variability and are distinct from the test-set bootstrap standard errors in Table 2.

Table 8: **Specialization-seed variability on WavLM-large.** Cells report min–max clean / pooled robust accuracy (%, ↑) across three independent specialization seeds.

Method	KWS	IC	SID-250	ER
θ_k	97.4–98.0 / 17.4–19.5	98.9–99.2 / 16.5–18.4	90.7–91.8 / 2.6–3.9	67.1–69.3 / 4.4–6.1
Ours (install + realign)	94.9–95.6 / 80.7–83.2	98.0–98.6 / 86.8–88.7	83.6–84.8 / 39.0–42.8	62.4–64.2 / 35.0–38.9

C.2 ASR READOUT REALIGNMENT

For CTC-based ASR, the unpatched descendant produces the greedy frame path $\hat{c}_t = \arg\max_v [W_0 h_{\theta_k,t}(x) + b_0]_v$ and pseudo-transcript $\hat{y}^{\text{ctc}}(x) = \mathcal{B}(\hat{c}_{1:T})$, where $\mathcal{B}$ is the standard CTC collapse operator. The prediction-matching term is the CTC loss between the realigned model's output and $\hat{y}^{\text{ctc}}(x)$. Because greedy CTC paths are dominated by blanks, applying the auxiliary margin uniformly over all frames would primarily encourage blank separation. We therefore average the margin hinge loss only over frames assigned a nonblank token by the original descendant, $\mathcal{T}_+(x) = \{t : \hat{c}_t \neq \varnothing\}$:

$$\mathcal{L}_{\text{ASR}} = \mathcal{L}_{\text{CTC}}(W h_{\theta_{k,\alpha}}(x) + b_0, \hat{y}^{\text{ctc}}(x)) + \frac{\beta}{|\mathcal{T}_+(x)|} \sum_{t \in \mathcal{T}_+(x)} \max\big(0,\ \gamma - m_{\hat{c}_t}(h_{\theta_{k,\alpha},t}(x); W, b_0)\big)\,, \tag{9}$$

with the second term omitted when $\mathcal{T}_+(x)$ is empty. All time steps, including those assigned blank by the original descendant, still participate in the sequence-level CTC prediction-matching term. The encoder remains frozen; only W is updated and $b = b_0$ is fixed. The auxiliary margin is applied at the frame-level CTC projection and is not treated as a sequence-level robustness guarantee.

D ADDITIONAL SPECIALIZATION REGIME: LORA

Main experiments use full encoder fine-tuning, which permits broader changes to the pretrained encoder than parameter-efficient adaptation. To evaluate a more constrained specialization regime, we obtain WavLM-large descendants with rank-16 LoRA adapters (Hu et al., 2022) on the attention query and value projections and the feed-forward FC1 and FC2 matrices. We use an effective LoRA scaling factor of 2 and the same specialization optimization settings as in Table 7. We retain the learned LoRA adapters, add the same robustness update to the underlying encoder at fixed $\alpha = 1$ without LoRA-specific tuning, and use the same label-free readout realignment procedure.

Table 9 shows that installation improves pooled robust accuracy by 51.9 percentage points on average, with a mean clean-accuracy loss of 2.2 points relative to the original LoRA descendants. The patch-only clean-accuracy loss is smaller under LoRA than under full fine-tuning (4.7 versus 13.0 points), while the full method yields comparable robust-accuracy gains across the two specialization regimes.

Table 9: **LoRA-specialized descendants, primary backbone.** Clean / pooled robust accuracy (%, ↑) on WavLM-large.

Configuration	KWS	IC	SID-250	ER
θ_k (LoRA)	96.4 / 17.5	98.8 / 17.2	84.2 / 2.1	65.7 / 5.7
Patch only	94.9 / 81.9	95.6 / 83.8	76.0 / 35.7	59.9 / 34.0
Ours (install + realign)	95.6 / 84.0	98.0 / 88.7	80.1 / 39.6	62.5 / 37.8

E THREAT MODEL AND ROBUSTNESS EVALUATION

This appendix gives the common evaluation protocol, attack-specific settings and feasibility checks, the perceptual operating-point study, and a targeted-ASR scope check supporting the threat model in Section 3.1.

E.1 ATTACK PROTOCOL AND SETTINGS

Waveform attack protocol. All seven attacks operate directly on the input waveform under the SNR-constrained threat set in Eq. 1, with $S = 37$ dB. The perturbation radius is computed separately for each utterance, and optimization enforces the ℓ_2 budget together with $x + \delta \in [-1, 1]^n$. Each attack targets the exact model being evaluated, including the installed update and realigned readout where applicable. All retained perturbations are verified against Eq. 1.

Standard attacks. For classification, AutoPGD uses untargeted cross-entropy with 200 steps, two independent runs, and its adaptive step-size schedule. C&W uses an untargeted logit-margin objective for classification, optimized with AdamW at learning rate 5×10^{-4} for 200 steps and two runs. Both are adapted to waveform inputs under the utterance-specific ℓ_2 constraint. SPSA uses cross-entropy for classification and CTC loss for ASR, with two-sided finite-difference gradient estimation, 32 iterations, 32 gradient samples per iteration, and two runs. This corresponds to 4,096 objective evaluations for gradient estimation per utterance; additional candidate-selection evaluations are not included in this count. The psychoacoustic attack is described in Section E.2.

Mechanism-aware attacks. For classification, the representation-drift attack maximizes $\|h_\theta(x + \delta) - h_\theta(x)\|_2$, while the margin attack minimizes the true-class geometric margin $m_y(h_\theta(x + \delta); W, b)$ in Eq. 3. The joint attack maximizes

$$\mathcal{L}_{\text{joint}}(x, \delta) = \|h_\theta(x + \delta) - h_\theta(x)\|_2 - m_y(h_\theta(x + \delta); W, b). \tag{10}$$

All three use projected gradient ascent with normalized gradients, 200 steps, two independent runs, and step size $0.2\epsilon_x$, where $\epsilon_x = \|x\|_2 10^{-37/20}$.

Evaluation and aggregation. Each attack retains its strongest feasible candidate across iterations and runs. For classification, an example is counted as robust only if all seven attacks fail. The minimum-margin selector in Eq. 5 is used separately for the representation and margin diagnostics and does not alter pooled robust accuracy. For label-dependent objectives, classification attacks use the ground-truth class and untargeted CTC attacks use the ground-truth transcript, rather than the pseudo-labels or pseudo-transcripts used during Stage 2 readout realignment. The targeted ASR experiment instead uses the specified target transcript. AutoPGD, C&W, the ASR counterpart of the margin attack, and SPSA use the sequence-level CTC loss against the reference transcript. The representation-drift attack remains frame-level, and the joint attack combines frame-level representation drift with the CTC objective. The highest-WER valid candidate is selected per utterance before computing corpus-level adversarial WER. Untargeted ASR results appear in Table 2; targeted success is reported separately in Table 11.

E.2 PERCEPTUAL OPERATING POINT AND PSYCHOACOUSTIC ATTACK

Perceptual operating point. Table 10 characterizes the 37 dB operating point with objective speech-quality and intelligibility metrics and a small-scale listening study. For each of six SNR budgets, we generated untargeted AutoPGD perturbations on 100 Common Voice English utterances. Leading and trailing nonspeech were removed *before* attack generation, while natural pauses within the utterances were retained. Twelve listeners completed 50 standard ABX trials each (600 trials in total). We report the ABX rates descriptively and make no population-level inference from this limited listener sample. At 37 dB, PESQ is 3.97, STOI is 0.998, and ABX accuracy is 65%, compared with 78% at 34 dB. Thus, the chosen budget preserves high measured intelligibility under this protocol, but does not make the perturbations inaudible. The study measures discrimination on trimmed clips containing speech, not on untrimmed recordings containing extended leading or trailing silence. The corresponding robustness-budget sweep is in Fig. 4. These examples use the standard SNR constraint; they are not samples from the separately masking-constrained psychoacoustic attack.

Table 10: **Speech-present perceptual evaluation across SNR budgets.** PESQ, STOI, and descriptive ABX correct-response rates for untargeted AutoPGD perturbations on 100 trimmed Common Voice English utterances per budget. Twelve listeners completed 50 ABX trials each (600 trials in total); chance performance is 50%. The 37 dB operating point is highlighted.

SNR budget (dB)	PESQ ↑	STOI ↑	ABX correct (%)
43	4.19	0.999	56
40	4.10	0.999	59
37 *(operating point)*	**3.97**	**0.998**	**65**
34	3.73	0.997	78
31	3.36	0.996	86
28	2.98	0.992	93

SNR-constrained psychoacoustic attack. The psychoacoustic attack in the seven-attack pool adapts the auditory-masking approach of Qin et al. (2019) to the untargeted downstream task objective. It targets the exact model being evaluated, including the installed update and realigned readout where present. In addition to the waveform constraint in Eq. 1, its perturbations satisfy a time–frequency masking bound:

$$\Delta_{S,\Psi}(x) = \big\{\delta \in \Delta_S(x) : |\hat{\delta}(f,t)|^2 \le \Psi_x(f,t)\ \forall f,t\big\}, \tag{11}$$

where $\hat{\delta}$ is the short-time Fourier transform of the perturbation and Ψ_x is the masking threshold computed from the clean waveform. The SNR constraint is a minimum of 37 dB: perturbations are not rescaled to use the full waveform budget if doing so would violate masking. Retained candidates must satisfy both Eq. 1 and the masking bound in Eq. 11; satisfying a masking penalty alone is insufficient. The attack is included for KWS, IC, SID-250, ER, and ASR in the same seven-attack pool as the other attacks. Because $\Delta_{S,\Psi}(x) \subseteq \Delta_S(x)$, this attack strengthens the evaluation with a masking-constrained adversary while preserving the headline SNR threat model.

E.3 Targeted ASR scope check

Targeted attack protocol. We evaluate targeted AutoPGD separately from the untargeted seven-attack pool, using 1,500 steps and two independent runs under the same 37 dB waveform constraint in Eq. 1. The fixed target transcript is "your model has been attacked." We select 500 Common Voice English utterances of duration 1–5 s whose reference transcripts contain none of the five target words. Each attack targets the exact model being evaluated. Sentence success is the percentage of decoded outputs matching the complete target sentence. Target-substring presence is the mean fraction of the five target-word strings found anywhere in the decoded output, counting each at most once per utterance; order is ignored and matches inside longer words count. This descriptive string-overlap metric is not word-aligned recognition: the source filter excludes the five whole target words but not their substrings. Table 11 reports both metrics as measures of progress toward the fixed target transcript. This Common Voice experiment is separate from the headline LibriSpeech adversarial-WER evaluation.

Table 11: **Targeted ASR scope check at 37 dB.** Targeted AutoPGD on 500 Common Voice English utterances with fixed target "your model has been attacked." Sentence success requires the complete decoded target; target-substring presence counts target-word strings anywhere in the output. Lower is better.

Method	Sentence success (%) ↓	Target-substring presence (%) ↓
θ_k	52.6	88.5
Ours (install + realign)	3.2	6.1
Task-specific AT	2.8	4.9

F Amortized compute

All timings are measured on one RTX 4090 and averaged over the evaluated configurations. Stage 1 averages 19.1 GPU-h across the three backbones and is run once per backbone. Across the 15

backbone–task descendants, Stage 2 averages 0.12 GPU-h per descendant, including pseudo-label generation, one-time representation caching, and readout realignment; task-specific AT averages 8.7 GPU-h per descendant for four epochs from θ_b. Thus, for K descendants, the average-cost model is $19.1 + 0.12K$ GPU-h for installation versus $8.7K$ GPU-h for task-specific AT, breaking even at $K \approx 2.2$ and requiring 16% of the corresponding AT compute at $K = 15$ (Table 12).

Table 12: **Amortized compute,** measured on one RTX 4090 (GPU-h). Stage 1 averages 19.1 across the three backbones and is paid once per backbone. Across all 15 backbone–task descendants, Stage 2 averages 0.12 and includes pseudo-label generation, one-time representation caching, and readout realignment; task-specific AT averages 8.7 for four epochs from θ_b. Training-time attacks are included and evaluation attacks excluded.

Route	Cost model	K=1	K=2	K=5	K=10	K=15
Task-specific AT *(reference)*	$8.7K$	8.7	17.4	43.5	87.0	130.5
Ours (install + realign)	$19.1 + 0.12K$	19.2	19.3	19.7	20.3	20.9
Ours / AT (%)	–	221	111	45	23	16

G Representation drift and descendant characterization

G.1 Drift measurement and scope

Section 4.1 defines the pool-aligned drift diagnostic through the selector in Eq. 5. After all seven attacks have returned valid candidates, the selector retains the candidate with the smallest ground-truth geometric margin; ties use a fixed attack order. This post-hoc choice is used only to obtain a single perturbation for the representation and margin diagnostics. Pooled robust accuracy itself is unchanged and still requires every attack in the pool to fail.

For classification, drift is the raw ℓ_2 distance between the final-layer pooled clean and perturbed representations. For an utterance with T frames, let $h_{\theta,t}(x)$ denote the final-layer representation at frame t and define $r_t = h_{\theta,t}(x+\delta) - h_{\theta,t}(x)$. Because classification uses mean pooling,

$$\left\| \frac{1}{T} \sum_{t=1}^{T} r_t \right\|_2 \leq \left(\frac{1}{T} \sum_{t=1}^{T} \|r_t\|_2^2 \right)^{1/2}. \tag{12}$$

Thus, pooled drift is upper-bounded by the root-mean-square frame drift underlying the Stage 1 stability term. No additional representation normalization is applied, so drift and the geometric margin in Fig. 3 have the same representation-space units. Readout realignment does not change the encoder response to a fixed perturbation. However, attacks are regenerated against each readout and the selector in Eq. 5 is readout-dependent, so reported mean drifts may differ. The resulting margin condition is sufficient only for the returned candidates, not a certificate over the full threat set. For CTC-based ASR, the analogous margin acts at the frame-level projection, while sequence robustness is evaluated directly using adversarial WER.

G.2 Descendant characteristics

Table 13 characterizes several descendant properties relevant to transfer after specialization. The relative task-vector norm measures how far each encoder moves from θ_b in weight space, while $\cos(\Delta_{\text{rob}}, \tau_k)$ measures global alignment between the robustness and specialization updates. The cosines are near zero and vary little across tasks.

To characterize source-to-downstream shift without confounding it with specialization, we compute F_k at the pretrained backbone θ_b between the Stage 1 source and descendant calibration distributions of pooled representations. Let X_s and X_k denote these representations, with means μ_s and μ_k, and define $A = (X_s - \mu_s)/\sqrt{n_s - 1}$ and $B = (X_k - \mu_k)/\sqrt{n_k - 1}$. We compute the Gaussian Fréchet (2-Wasserstein) distance exactly in sample space as

$$F_k = \left(\|\mu_s - \mu_k\|_2^2 + \|A\|_F^2 + \|B\|_F^2 - 2\|BA^\top\|_* \right)^{1/2}, \tag{13}$$

where $\|\cdot\|_*$ is the nuclear norm. This is equivalent to the standard covariance square-root formulation but avoids taking a matrix square root of rank-deficient empirical covariances. The representation-normalized margin $\bar{m}_k$ is the original descendant's mean clean geometric margin divided by its mean representation norm, providing a scale-adjusted diagnostic; AT headroom is taken from Table 2.

KWS and IC have smaller task-vector norms and source-to-downstream Fréchet distances and larger absolute robustness gains than SID-250 and ER. However, the orderings of F_k, normalized margin, and $\Delta\mathrm{RA}_k$ are not uniformly monotonic, and gains should also be interpreted relative to the gap to task-specific AT. With four tasks, we treat Table 13 descriptively and do not fit predictive models or infer causality.

Table 13: **Descendant characteristics and robustness gain, primary backbone.** Characteristics are measured for the four WavLM-large classification descendants; $\Delta\mathrm{RA}_k$ is the gain from θ_k to the full method. The four tasks are treated descriptively rather than for fitting a predictive model.

Descendant property	KWS	IC	SID-250	ER
Specialization geometry				
$\|\tau_k\|_2/\|\theta_b\|_2$, relative task-vector size	0.0102	0.0092	0.0148	0.0170
$\cos(\Delta_{\mathrm{rob}}, \tau_k)$, update vs. task vector	0.0035	0.0042	0.0029	0.0076
Source-to-downstream shift				
F_k, Fréchet distance at θ_b	13.8	15.3	30.1	25.6
Downstream robustness context				
AT headroom $\mathrm{RA}_{\mathrm{AT}} - \mathrm{RA}_{\theta_k}$ (points)	71.8	76.3	53.6	43.3
$\bar{m}_k$, representation-normalized margin ↑	1.68	1.05	0.59	0.86
Outcome				
$\Delta\mathrm{RA}_k$, robustness gain (points) ↑	63.8	70.1	37.7	31.8

H REALIGNMENT DATA EFFICIENCY

Table 14 reports clean and pooled robust accuracy as the amount of clean, unlabeled calibration audio used for readout realignment is reduced from the headline 45-minute budget. Across the four classification descendants, the 5-, 15-, and 30-minute conditions draw from the same 45-minute pool; the installed encoder, objective, and optimization settings are unchanged. For SID-250, we report only 30 and 45 minutes because the shorter budgets are too small to construct subsets covering all 250 speakers; including them would confound calibration duration with speaker coverage. Speaker identities are used only to ensure comparable subsets; the realignment objective remains label-free.

KWS is essentially saturated by 15 minutes and IC by 30 minutes, whereas ER and SID-250 continue to improve with more calibration audio. We retain 45 minutes as a common headline budget rather than tuning calibration duration per task.

Table 14: **Realignment data efficiency, primary backbone.** Cells report clean / pooled robust accuracy (%, ↑) as clean unlabeled calibration audio is reduced from the 45 min headline budget. Smaller conditions use subsets of the same disjoint 45-minute holdout; all other Stage 2 settings are fixed.

Task	5 min	15 min	30 min	45 min
KWS	90.0 / 78.7	94.1 / 82.6	95.3 / 82.4	95.2 / 82.1
IC	83.2 / 74.3	92.5 / 83.7	97.6 / 87.9	98.2 / 87.6
SID-250	–	–	80.6 / 38.1	84.1 / 40.9
ER	49.8 / 27.2	56.7 / 30.8	61.0 / 35.7	63.4 / 37.1