ADAPTATION OR RECALIBRATION? CONTINUAL TEST-TIME ADAPTATION FOR AUDIO DEEPFAKE DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Audio deepfake detectors meet new attacks after deployment: a detector trained on today's synthesis systems must run against generators and recording conditions that are not known in advance. We study this as source-free, label-free continual test-time adaptation (CTTA) and build a controlled, strictly prequential benchmark for audio deepfake detection, in which every utterance is scored before it can affect an update. The benchmark has five streams and $266{,}400$ scored positions, with source training on ASVspoof 2019 LA and target streams pairing MLAAD spoofed speech with M-AILABS bona fide speech. It separates two failures that a single accuracy number hides: discrimination fails when the score ordering gets worse, and the operating point fails when the scores move relative to a threshold fixed before deployment. Five prediction-driven CTTA methods, evaluated on XLS-R and WavLM, move the operating point far more than they repair discrimination. On the stream of 52 sequential generators, Tent lowers per-block minimum detection cost (minDCF) by 0.010 and 0.013 on the two held-out checkpoints, against 0.100–0.112 for a labeled update of the same parameters. Recalibrating the frozen detector removes 22–77% of its calibration loss, depending on the checkpoint, without changing the ordering. An utterance's own score cannot say which of two utterances is ranked wrongly; relations between utterances can. We introduce CAST (Consensus-Anchored Score Transfer), which smooths frozen scores over reciprocal nearest neighbors and distills the resulting order into the adapting detector with a pairwise ranking loss. On the two held-out checkpoints of that stream, CAST lowers minDCF by 0.065 adapting only the LayerNorm affine parameters and by 0.103 adapting the full transformer encoder, where EER reaches 11.3%, against 0.011 for the strongest baseline. The gain holds on all three controlled target streams and both backbones. Smoothed scores improve discrimination even with no parameter update, while CAST's loss reads only score differences and leaves the operating point unresolved. Label-free continual detection therefore needs to solve two problems, and repairing discrimination solves only one.

1 INTRODUCTION

Audio deepfake detectors are trained on a finite set of attacks and then deployed against generators and recording conditions they have never seen in training. MLAAD alone contains 175 text-to-speech models in 54 languages (Müller et al., 2024b). Existing work handles this mismatch under different assumptions about what becomes available after training. Few-shot methods need a small labeled sample of the new attack (Kukanov et al., 2024). Domain-generalization methods do not need target data, but do not adapt to new conditions (Laakkonen et al., 2026). Continual UDA adapts across unlabeled target domains, but keeps labeled source data during training (Chen et al., 2025a), and recent MLOps work collects new adaptation data once drift is detected (Wang et al., 2026).

All of these assumptions can fail at once. A forensic laboratory or a fraud-screening service may keep no source corpus on the deployed machine, may have no ground truth for the audio it receives, and must return a decision on each recording when it arrives rather than after the next retraining

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.

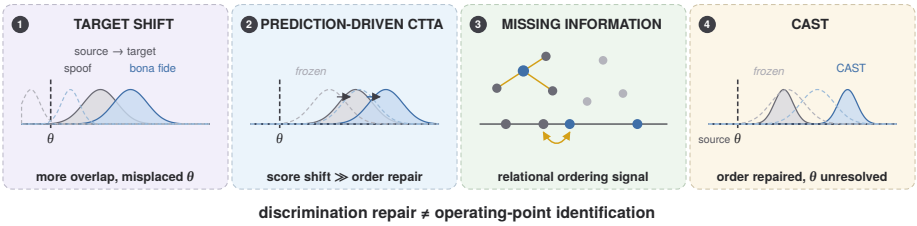

Figure 1: **Paper overview. Score densities are schematic.** Faded dashed curves denote the reference state. (1) Shift can worsen the score ordering and move the scores relative to the source threshold θ, producing discrimination loss and calibration loss. (2) Prediction-driven CTTA mostly does the second. (3) An utterance's own score cannot show which pairs should swap; relations between utterances can. (4) CAST distills those relations into an online ranking update. Figure 2 quantifies (2); Figure 3 details CAST.

cycle. Nothing tells the system when the attack distribution has changed. We study this regime as source-free, label-free continual test-time adaptation (CTTA) for audio deepfake detection (Wang et al., 2021; 2022; Liang et al., 2025): no source data, no target labels, no domain boundaries, and every utterance scored before it can affect an update.

Such a detector also uses a threshold that was fixed before the target stream began. Distribution shift can then cause two different failures. It can worsen the score ordering, which lowers discrimination. It can also move the scores relative to the fixed threshold, which moves the operating point while the ordering stays intact. Accuracy cannot separate the two, because it evaluates decisions at one implicit threshold (Wang et al., 2022; Press et al., 2023). We therefore use the actual-minus-minimum decomposition of detection cost (Brümmer & du Preez, 2006), which splits delivered cost into discrimination and calibration loss (Section 3). This gives our main question: *what can label-free continual adaptation repair, discrimination or the operating point?*

Our benchmark trains on ASVspoof 2019 LA (Todisco et al., 2019; Wang et al., 2020) and pairs spoofed speech from MLAAD (Müller et al., 2024b) with bonafide speech from M-AILABS (Solak, 2019). Its five streams and $266{,}400$ scored positions cover source retention, a fixed target shift, a generator sequence, a class-prior shift, and locale breadth.

Prediction-driven CTTA moves the operating point far more than it repairs discrimination. This holds for five methods on XLS-R and on WavLM. On GENERATOR, Tent lowers per-block minDCF by 0.010 and 0.013 on the two held-out checkpoints, against 0.100–0.112 for a labeled update of the same parameters. Recalibrating the frozen detector lowers calibration loss on its own, without touching the ordering.

Correcting an ordering means knowing which utterances should move past which others. A single prediction does not provide that information; it only places an utterance in the detector's current ordering. Any non-decreasing transformation of that score preserves or coarsens the same ordering. Relations between utterances can provide the missing information. We introduce CAST, which smooths frozen scores over reciprocal nearest neighbors and distills the resulting order into the adapting detector with a pairwise ranking loss. It uses no class pseudo-label, no target threshold, and no class-prior estimate. CAST improves discrimination on all three controlled target streams and on both backbones, and the smoothed scores improve it even with no adaptation. The operating point remains open. CAST's loss depends only on score differences, so a better ordering does not tell the detector where to put its threshold. Repairing discrimination and finding the operating point are separate problems, and we solve only the first.

2 Background and Related Work

Prediction-driven test-time adaptation. Test-time adaptation (TTA) updates a trained model on the target data it sees at inference time. We work in the source-free, label-free continual setting, where source data and target labels are unavailable and the target distribution changes over time (Liang et al., 2025). The methods we evaluate all read the model's own predictions. Tent (Wang et al., 2021)

minimizes prediction entropy on the normalization affine parameters. EATA (Niu et al., 2022) keeps only confident, non-redundant samples and anchors the weights to their source values. SAR (Niu et al., 2023) removes high-gradient samples and seeks a flat minimum, for stability under small or imbalanced batches. RoTTA (Yuan et al., 2023) replaces the batch statistics with running estimates and holds a class-balanced memory. RDumb (Press et al., 2023) resets the model at a fixed period, on the argument that long adaptation runs collapse. Other work in the same family filters or reweights layers and gradients (Sahoo et al., 2025; Chen et al., 2025c); see Liang et al. (2025) for a full account.

The prediction channel is also where recent theory locates the limits. Goyal et al. (2022) derive the meta-optimal test-time loss from the training loss, and for a cross-entropy-trained classifier that loss is temperature-scaled entropy. Our source models are cross-entropy trained, so Tent with one temperature is the conjugate method here, not a heuristic alternative to it. Related work asks when a proxy gradient aligns with the task gradient (Zhou et al., 2026) and when unlabeled evidence identifies the correct update at all (Jhawar & Wang, 2026). We ask the detection version of that question: does a prediction reveal which score orderings should change?

Neighborhood structure at test time. Nearest-neighbor structure can carry information beyond a model's own prediction. Prior TTA methods turn neighbors into class targets or contrastive objectives (Jang et al., 2023; Yang et al., 2021; Chen et al., 2022a). CAST instead turns frozen neighborhood relations into pairwise ordering constraints, and needs neither a target threshold nor a class-prior estimate. The distinction matters for detection: a class target can be met by shifting every score toward that class, whereas a loss on score differences cannot. Section 6 tests the class-level alternative on the same neighborhood signal.

Evaluation and recalibration. TTA results are sensitive to batch size, class balance, and hyperparameter transfer (Zhao et al., 2023), and source retention is treated explicitly by EATA (Niu et al., 2022) and CoTTA (Wang et al., 2022). Separately, target class proportions can be estimated from model outputs and used to recalibrate decisions without changing the representation (Saerens et al., 2002; Lipton et al., 2018). What these evaluations share is a single implicit threshold: accuracy scores every decision at one operating point, so a method that repairs the ordering and a method that moves the scores produce the same summary number. Detection has a standard tool for splitting them. The actual-minus-minimum decomposition separates threshold-optimized discrimination from the extra cost of the deployed operating point (Brümmer & du Preez, 2006; Kinnunen et al., 2018), and we use it throughout.

Audio deepfake detection under evolving shift. Generalization to unseen generators remains a central problem in audio deepfake detection (Müller et al., 2022; 2024a), and corpus artifacts such as leading silence can substitute for it (Müller et al., 2021). Existing approaches differ in what they assume becomes available after source training. Few-shot and supervised test-time adaptation use labeled target examples (Kukanov et al., 2024; Oiso et al., 2024). Domain generalization seeks robustness with no target adaptation at all (Laakkonen et al., 2026). Supervised continual learning updates the detector as new attacks arrive (Ma et al., 2021; Zhang et al., 2024; Chen et al., 2025b).

Continual UDA assumes less than any of these. Chen et al. (2025a) adapt across unlabeled target domains in sequence, but keep labeled source data for domain-adversarial training and treat each target domain as an explicit stage. Wang et al. (2026) monitor drift from new TTS attacks and fine-tune the detector on similarly drifted data. One still needs the source corpus, the other still needs new data to be collected before it can act. We evaluate strictly prequential CTTA, in which no source data, target labels, or domain boundaries are available, at a threshold fixed on the source data, and report discrimination and calibration loss separately. We are not aware of previous work on audio deepfake detection in this regime.

3 A Controlled Continual Stream Benchmark

A deployed detector scores each utterance at a threshold fixed on source data, and has no target labels with which to retune it. The benchmark evaluates cost at that fixed threshold and splits it into discrimination and calibration loss. We trained the source models on ASVspoof 2019 LA (Todisco et al., 2019; Wang et al., 2020). The target streams pair spoofed speech from MLAAD (Müller et al., 2024b) with bonafide speech from M-AILABS (Solak, 2019), the corpus MLAAD was synthesized from. Corpus identity therefore predicts the class, and a detector could exploit that instead of the

Table 1: Benchmark streams. SOURCE is the source-regime control. STATIONARY, GENERATOR, and PRIOR support the main target claims; LOCALE is a breadth check. R is the spoofed fraction. A *block* is a contiguous segment in which the varying factor is constant. Single-block streams are reported over 600-utterance windows.

Stream	Varying factor	Structure	R	Purpose
SOURCE	none	$1 \times 70{,}800$	≈ 0.90	source retention
STATIONARY	none	$1 \times 63{,}600$	0.50	fixed target shift
GENERATOR	generator	$52 \times 1{,}200$	0.50	generator sequence
PRIOR	class prior	$5 \times 12{,}000$	0.05–0.50	prior shift and recovery
LOCALE	locale / recording	$8 \times 1{,}200$	0.50	locale breadth

spoofing artifacts. We treat this as the main threat to the benchmark and address it under *Corpus controls* below.

Protocol. The benchmark is strictly prequential (Gama et al., 2013). Each batch of $B = 16$ utterances is scored and recorded before any update, and a recorded score is never revised. A method receives no target labels, source data, block boundaries, stream position, or future utterances (Liang et al., 2020; Wang et al., 2021). Every stream starts from the same source checkpoint, and nothing resets the model at a block boundary. Appendix A.1 gives the full protocol.

We call the unadapted source detector *frozen*. The main detector combines XLS-R (Babu et al., 2022) with AASIST (Jung et al., 2022; Tak et al., 2022), and we repeat the three main target streams with WavLM Large (Chen et al., 2022b) for CAST-LN, all five prediction-driven baselines, and the matched labeled update. We trained three checkpoints per backbone and selected each one by minimum validation negative log-likelihood, a rule we declared before any evaluation and applied identically to both backbones. The deployed threshold θ is the equal-error-rate (EER) threshold on the ASVspoof 2019 LA development set, computed per checkpoint and never retuned on target data. The main conclusions also hold at the source-development minDCF threshold (Appendix A.4).

Streams. Table 1 summarizes the five streams. Model state carries across blocks, so a later block includes the effect of earlier updates, and blockwise trends are descriptive rather than isolated causal effects. GENERATOR contains 52 English MLAAD generators. We selected them by the frozen detector's minimum half total error rate (min-HTER), the minimum of $(\mathrm{FAR} + \mathrm{FRR})/2$ over all thresholds, using a band declared in advance; Appendix A.3 gives the inclusion rule and the ordering. PRIOR runs five blocks with spoofed fractions $R = 0.50 \to 0.20 \to 0.05 \to 0.20 \to 0.50$. LOCALE varies locale and recording condition together and serves only as a breadth check.

Delivered cost and discrimination. Accuracy scores every decision at one implicit threshold, so it cannot separate a change in the score ordering from a change in score location (Wang et al., 2022; Press et al., 2023). We use the normalized ASVspoof 5 Track-1 detection cost instead, with $C_{\mathrm{miss}} = 1$, $C_{\mathrm{fa}} = 10$, and $\pi_{\mathrm{spf}} = 0.05$ (Wang et al., 2024; Kinnunen et al., 2018). The application prior $\pi_{\mathrm{spf}} = 0.05$ is fixed, independently of the empirical spoofed fraction R, so PRIOR changes the observed class prevalence while the cost function stays the same. bonafide speech is the target class, so P_{miss} is the false-rejection rate (FRR) and P_{fa} the false-acceptance rate (FAR). After normalization,

$$\mathrm{DCF}(\tau) = 1.90 P_{\mathrm{miss}}(\tau) + P_{\mathrm{fa}}(\tau).$$

Let actDCF be the cost at the deployed threshold θ, and minDCF its minimum over all thresholds. Following Brümmer & du Preez (2006),

$$\underbrace{\mathrm{actDCF}}_{\text{delivered cost}} = \underbrace{\mathrm{minDCF}}_{\text{discrimination}} + \underbrace{(\mathrm{actDCF} - \mathrm{minDCF})}_{\text{calibration loss}}. \tag{1}$$

Moving the threshold leaves the score ordering untouched, so minDCF measures discrimination and $\mathrm{actDCF} - \mathrm{minDCF}$ measures the cost of the operating point. Here *calibration* means where the operating point sits, not probabilistic calibration (Guo et al., 2017). We also report FAR and FRR.

Aggregation and validity. Adaptation can change the score scale over time, so we compute threshold-optimized metrics locally: by block on GENERATOR and PRIOR, and over consecutive

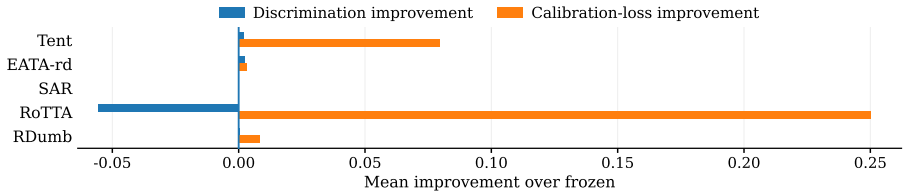

Figure 2: **What prediction-driven CTTA improves.** Mean change from the frozen detector across the three main XLS-R target streams. Discrimination improvement is $-\Delta\mathrm{minDCF}$ and calibration-loss improvement is $-\Delta(\mathrm{actDCF} - \mathrm{minDCF})$; positive values are better.

600-utterance windows on SOURCE and STATIONARY. Inferential claims use within-checkpoint paired differences from the frozen detector, with bootstrap intervals stratified by class and by block or window (Efron & Tibshirani, 1993; Bisani & Ney, 2004). We never pool scores across checkpoints. Appendix A.5 gives the aggregation and resolution checks.

Corpus controls. Because the two classes come from different corpora, we removed the cues that separate those corpora before building any stream, and audited what survived. Both corpora carry low-level leading and trailing silence, which is itself predictive of the class, so we applied energy-based voice activity detection to every partition, including the source training data. MLAAD ships at 22.05 kHz and M-AILABS at 16 kHz, and the resampling step leaves a bandwidth signature that classifies the two corpora almost perfectly, so we low-pass filtered every recording below 7.7 kHz at its native rate before resampling. Neither control is standard for this corpus pair, and we adopted both on our own measurements rather than on precedent. We applied both once, at dataset construction, and recorded them in the stored data. We then audited sample rate, edge silence, signal level, duration, and padding behavior (Geirhos et al., 2020; Müller et al., 2021; 2022). After these controls the strongest remaining cue reaches 42.7–42.9% EER, against 50% for a cue carrying no information (Appendix A.6).

4 DIAGNOSING PREDICTION-DRIVEN LABEL-FREE ADAPTATION

Prediction-driven CTTA moves the operating point far more than it repairs discrimination, on both backbones. We show this against two references: a label-free recalibration of the frozen detector, and a labeled update of the same parameters.

4.1 SETUP, BASELINES AND LABELED REFERENCE

We evaluate the five baselines described in Section 2: Tent (Wang et al., 2021), EATA (Niu et al., 2022), SAR (Niu et al., 2023), RoTTA (Yuan et al., 2023), and RDumb (Press et al., 2023). All of them update the same set of normalization-layer affine parameters. EATA is run without its Fisher regularizer, since that term needs source data, which our setting does not allow; we write EATA-rd for this variant. Appendix B gives the adapted parameter sets, the reference-arm definitions, and checks on weak baseline movement.

MLLS (Saerens et al., 2002) estimates the recent spoofed fraction and recalibrates the frozen detector, leaving its parameters untouched. The labeled update replaces Tent's entropy loss with cross-entropy against the true labels, and keeps its parameter set, optimizer, learning rate, batch size, and update order. It is not deployable, and measures how much discrimination the same parameters can gain when the labels are correct. Neither reference is a baseline: both show what is available when one of the two failures is addressed with information the deployed setting does not have.

4.2 DISCRIMINATION VERSUS CALIBRATION

In the three main target streams, the five baselines barely move discrimination, while their changes in calibration loss are several times larger (Figure 2). RoTTA separates the two most clearly: its discrimination gets worse on average, while its calibration loss improves.

On GENERATOR, Tent changes per-block minDCF by -0.003, -0.013, and -0.010 on the three XLS-R checkpoints, against -0.110, -0.112, and -0.100 for the labeled update of the same parameters. Correct labels therefore buy roughly ten times the discrimination gain on the same parameter set, and neither the parameter set nor the learning rate explains that gap, as Section 6 shows with a method that uses both. On STATIONARY, RoTTA lowers actDCF while raising minDCF: lower delivered cost does not imply better discrimination (Appendix F).

Calibration loss can also be reduced without any adaptation. MLLS removes 22–77% of it on XLS-R and 38–76% on WavLM, depending on the checkpoint, and leaves the score ordering exactly as it was. Its pure-label-shift assumption, $p_t(x \mid y) = p_s(x \mid y)$ (Lipton et al., 2018), does not hold across our corpus shift, which is the likely reason it recovers only part of the loss. Appendix F gives the full recalibration results and a temperature-scaling control.

4.3 AN UTTERANCE'S OWN SCORE CARRIES NO ORDERING INFORMATION

Any non-decreasing function of an utterance's scalar score keeps that utterance in the same position or merges it with others. Using it as a pseudo-score, this function , therefore, cannot lower minDCF. Thresholded pseudo-labels (Lee, 2013) merge positions in the same way, and entropy and confidence discard the class direction (Lee et al., 2024). Repairing an ordering needs information from somewhere other than the utterance's own score. This is a statement about what a pseudo-score contains, not an impossibility result for gradient adaptation: a parameter update changes the scores, and with them the ordering. The measured gap points the same way, though, as the baselines in Section 4.2 show.

We therefore screened candidate signals as scorers, with no adaptation at all. Own-score pseudo-labels do not improve per-block minDCF. Averaging frozen scores over reciprocal nearest neighbors in the frozen representation does, on both held-out checkpoints and all three main target streams (Appendix D.1). These neighborhoods are also informative where the detector is wrong: on checkpoint 1234, they hold about 91% label purity for the utterances the frozen detector misclassifies at θ, against 50% after permuting labels within the block. Section 5 builds CAST on these relations.

5 CAST: ADAPTATION FROM NEIGHBORHOOD ORDERING

Reciprocal-neighbor relations supply the ordering information an utterance's own score lacks. CAST smooths the frozen scores over those neighbors, orders the current batch by the result, and trains the adapting detector to reproduce that order. It assigns no class pseudo-label and estimates no target threshold or class prior. Appendix D.2 examines what the neighborhoods encode.

5.1 RECIPROCAL-NEIGHBOR ORDERING SIGNAL

A frozen copy of the source detector runs alongside the adapting one. For each utterance i it produces a detection score s_i and an embedding h_i, obtained by mean-pooling transformer encoder layer 4 over time. A first-in, first-out memory holds the embeddings and frozen scores of the most recent $N = 1{,}200$ utterances.

The search pool for each batch is the memory plus the current batch, so batch utterances can be neighbors of one another; no future utterance is available. We center the embeddings on the pool mean and ℓ_2-normalize them. For each batch utterance i, $\mathcal{N}_k(i)$ holds its $k = 10$ nearest neighbors under cosine similarity, excluding i. We keep only reciprocal neighbors: $j \in \mathcal{N}_k^{\mathrm{rec}}(i)$ if $j \in \mathcal{N}_k(i)$ and $i \in \mathcal{N}_k(j)$, a standard way to improve neighbor reliability (Qin et al., 2011; Zhong et al., 2017). The smoothed pseudo-score is

$$\tilde{s}_i = \frac{1}{1 + |\mathcal{N}_k^{\mathrm{rec}}(i)|} \left(s_i + \sum_{j \in \mathcal{N}_k^{\mathrm{rec}}(i)} s_j \right). \tag{2}$$

If i has no reciprocal neighbor, $\tilde{s}_i = s_i$. CAST uses $\tilde{s}_i$ only to define relative order, never as a target class; Section 6 evaluates the class-level alternative on the same signal. Appendix C reports the layer selection and the reciprocity ablation.

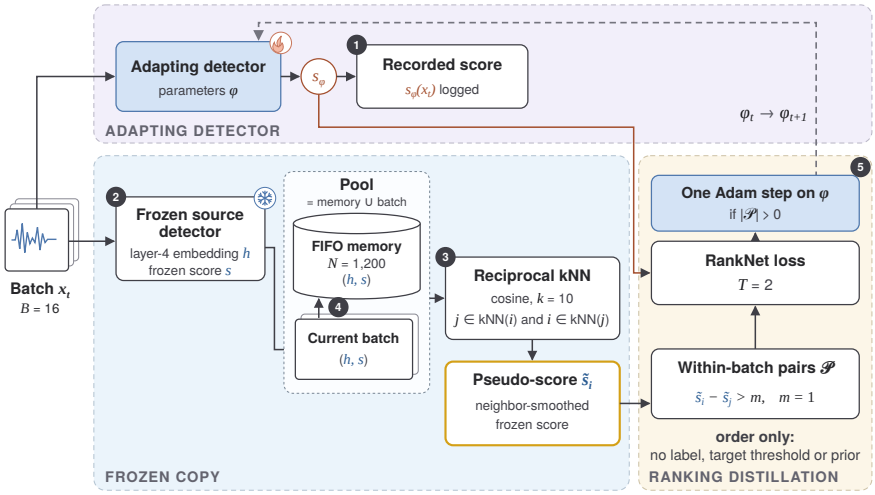

Figure 3: **CAST within one prequential batch.** (1) The adapting detector scores the batch before any update. (2) The frozen detector produces scores and embeddings. (3) Stored and current embeddings define reciprocal neighbors, which smooth the frozen scores into the pseudo-scores of equation 2. (4) The current scores and embeddings enter memory. (5) One RankNet update follows, if any pair survives the margin. The batch can therefore affect only later recorded scores.

5.2 Ranking distillation

For a batch of $B = 16$, the pair set $\mathcal{P}$ holds every ordered pair (i, j) with $\tilde{s}_i - \tilde{s}_j > m$. Pairs inside the margin, ties included, are discarded. Memory items shape the pseudo-scores but never enter $\mathcal{P}$. When $\mathcal{P}$ is empty, no update is taken. Otherwise we train the adapting detector to reproduce the pseudo-score ordering with the RankNet pairwise logistic loss (Burges et al., 2005):

$$\mathcal{L}_{\text{rank}} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \log\left(1 + \exp\left[-\frac{s_\phi(x_i) - s_\phi(x_j)}{T}\right]\right), \qquad m = 1,\ T = 2. \tag{3}$$

This is ranking distillation (Tang & Wang, 2018): the smoothed frozen scores give a target ordering, not target classes. The loss depends only on score differences, so adding the same constant to every score leaves it unchanged. CAST can learn a relative ordering, but nothing in the loss fixes where those scores sit relative to θ.

CAST-LN and CAST-FT are two adaptation surfaces for the same signal and the same loss, differing only in capacity. CAST-LN updates the 108,544 LayerNorm affine parameters of the transformer trunk, the set SAR also adapts, which makes it the like-for-like comparison with the baselines. CAST-FT updates the 311,227,520 transformer-trunk parameters and leaves the convolutional feature encoder and the AASIST back-end frozen. Both use Adam (Kingma & Ba, 2015). We selected the hyperparameters from a predeclared grid on one XLS-R checkpoint of GENERATOR and then fixed them. Target labels entered only that offline step: the other XLS-R checkpoints, every other stream, and all WavLM runs were held out from selection. Appendices C and C.4 give the optimization procedure, the selection procedure, and nearby parameter sets.

6 Results

We measure discrimination with within-checkpoint paired differences from the frozen detector. Checkpoint 1234 was used for CAST development on GENERATOR, so CAST results on that stream are reported on checkpoints 1235 and 1236 throughout, and so are the comparisons against the smoothed scorer and the class-target variants below. CAST improves discrimination far more than the prediction-driven baselines do, on all three main target streams. On the two held-out GENERATOR checkpoints, CAST-LN lowers minDCF by 0.065 and CAST-FT by 0.103, against 0.011 for Tent, the strongest baseline on that stream. STATIONARY and PRIOR show the same separation. Matched to Tent's parameter set and learning rate, CAST still changes minDCF by -0.021

Table 2: XLS-R results on all five streams. Δ is the change from the frozen detector in mean local minDCF, written without the leading zero; negative is better. EER is the corresponding mean local equal-error rate (%). Bold marks the best value in each column among the deployable rows; the labeled rows are excluded from that comparison. Values average three source checkpoints unless marked *, which averages checkpoints 1235 and 1236 and excludes the checkpoint used for CAST development on GENERATOR; a held-out comparison in the text can therefore differ from the table mean for the same entry. Within-checkpoint paired bootstrap intervals for the adaptation rows on the three main target streams are in Appendix F, together with the per-checkpoint values. SOURCE is the source-regime control. The labeled rows are not deployable. Labeled-FT has no SOURCE entry: the labeled trunk arm on SOURCE was run as a separate diagnostic on a different parameter set and is reported in Appendix F.

	SOURCE		STATIONARY		GENERATOR		PRIOR		LOCALE	
Method	Δ	EER	Δ	EER	Δ	EER	Δ	EER	Δ	EER
Frozen	–	6.3	–	15.7	–	14.3	–	15.7	–	17.0
Tent	$+.001$	6.2	$+.005$	16.2	$-.009$	14.1	$-.002$	16.0	$+.004$	17.3
EATA-rd	$-.000$	6.3	$-.001$	15.6	$-.003$	14.2	$-.003$	15.7	$-.001$	17.1
SAR	.000	6.3	.000	15.7	.000	14.3	.000	15.7	.000	17.0
RoTTA	$+.002$	6.4	$+.034$	16.8	$+.028$	15.5	$+.105$	19.8	$+.051$	17.6
RDumb	$-.001$	6.3	$-.001$	15.7	$+.002$	14.5	$-.002$	15.7	$+.004$	17.3
CAST-LN	$\mathbf{-.017}$	**5.6**	$-.069$	13.2	$-.065^*$	12.3^*	$-.021$	14.8	$-.008$	16.1
CAST-FT	$-.004$	6.7	$\mathbf{-.115}$	**11.8**	$\mathbf{-.103}^*$	$\mathbf{11.3}^*$	$\mathbf{-.098}$	**12.4**	$\mathbf{-.013}$	**15.7**
Labeled-LN	$-.012$	5.5	$-.100$	12.2	$-.108$	10.7	$-.080$	13.0	$-.010$	16.0
Labeled-FT	–	–	$-.398$	1.1	$-.387$	0.5	$-.407$	1.2	$-.149$	9.9

against Tent's -0.011, so the gap in Section 4.2 does not come from the adapted surface or the step size (Appendix C.4). Tuning Tent does not close it either: selecting among 16 configurations on checkpoint 1234 with the same offline statistic gives $\Delta\mathrm{minDCF} = +0.022$ on the held-out checkpoints, worse than the published configuration (Appendix B.4). Neither CAST configuration reaches its matched labeled reference. On WavLM the discrimination gain holds, but adapting on a target stream then degrades source performance, most clearly after GENERATOR and PRIOR; labeled replay removes that degradation and gives back part of the target gain (Appendix F).

Direct neighborhood smoothing. The smoothed pseudo-score can be used directly, with no parameter update. On the held-out checkpoints it improves discrimination more than CAST-LN on every XLS-R target stream: 0.100 against 0.065 on GENERATOR, and 0.091 against 0.023 on PRIOR, both pairs on those two checkpoints. It fails on the other two axes. Once the scorer and the frozen detector each receive their own label-free threshold, the scorer delivers lower cost on one of three XLS-R checkpoints per main target stream, and on none for LOCALE. On SOURCE it raises minDCF by 0.112 and worsens 117 of 118 windows, where CAST-LN lowers it by 0.017. On WavLM the recalibrated scorer beats the recalibrated frozen detector on two or three checkpoints depending on the stream; it was not run on WavLM SOURCE. Appendix E.1 gives the full results.

Class targets from the same neighborhoods. We also turn the same neighborhood signal into class targets, using the mean neighbor posterior or its thresholded version. On the held-out GENERATOR checkpoints the soft variant lowers minDCF by 0.031–0.041, against 0.062 and 0.067 for CAST-LN on the same two checkpoints; the hard variant raises it on both. Both drive the detector toward accepting almost every trial: FAR exceeds 92% in 17 of the 18 runs, that is two variants on three streams and three checkpoints. Neither delivers lower cost after label-free recalibration in any of the 18. Their updates are nevertheless aligned with the labeled reference at least as well as CAST's on the same parameters (Appendix E.2).

Both the neighborhood signal and its direction are necessary. Setting $k = 0$ removes the signal and raises minDCF by 0.007–0.017 on all three GENERATOR checkpoints. Reversing the target ordering raises it by 0.49–0.51 and pushes FAR above 99.6%. The gain also weakens under severe class imbalance: CAST-LN is worse than frozen in the 5%-spoof block of PRIOR (Appendix D.3).

Operating-point behavior. Better discrimination does not fix where the scores sit. CAST can improve the local ordering and invalidate the deployed threshold in the same run: on SOURCE, CAST-LN lowers local minDCF while FAR at the fixed threshold rises from about 17% to 84–89%, and CAST-FT reaches 96.8% FAR on checkpoint 1234 with minDCF still improved. A labeled source diagnostic moves FAR as far on two of three checkpoints, so this is not an artifact of the label-free objective.

We isolate the effect with a causal monotone transformation, estimated from past scores and held fixed within each 600-utterance window. It preserves the within-window ordering and minDCF, and changes delivered cost substantially. Restoring the frozen score location lowers actDCF by 0.60–0.79 on SOURCE and raises it by 0.02–0.39 on the three main target streams (Appendix E).

7 DISCUSSION AND LIMITATIONS

Our constructed benchmark clearly identifies where baseline CTTA methods fail work perform well. Label-free continual detection holds two problems, and the methods we tested solve at most one each: prediction-driven CTTA moves the operating point, CAST improves discrimination, and neither does both. The monotone transformation makes the separation concrete: local ordering can be preserved exactly while delivered cost moves, and restoring the source score location helps on SOURCE and hurts after target shift. This is not a property of CAST's ranking loss. Direct neighborhood smoothing and the class-target variants show the same split (Appendices E.1 and E.2).

The corpus split is our most serious limitation. Bonafide and spoofed target speech come from different corpora, so a detector could separate the corpora instead of the spoofing artifacts. We removed the cues we could measure (Section 3) and then asked how much of the effect survives removing corpus directions from the representation. Projecting out linearly decodable corpus directions retains 84% of the direct-smoothing gain on GENERATOR, but only 45% on STATIONARY and 52% on PRIOR. Matched random-direction removal does not cause a comparable loss, so the projection is removing something specific. Substantial k-NN corpus separability also remains afterwards. We therefore cannot claim that the gain on STATIONARY and PRIOR is independent of corpus identity, and a target corpus whose two classes share a recording pipeline is the test this benchmark cannot provide (Appendix D.2).

The neighborhood structure is speaker-dependent, and our probe cannot rule out that the gain depends on it. On the bonafide side, 99.9% of reciprocal neighbors share the speaker, against 26.4% for a pool-matched draw, because only four M-AILABS voices recur through the streams. On the spoofed side, nearest-neighbor retrieval of the source speaker is at or below chance, while generator identity reaches about 0.82. Holding out the recurring speaker does not settle the question: removing him also removes nearly all of the labeled update's headroom, so the hold-out establishes neither that CAST is speaker-invariant nor that it is not (Appendix D.2).

Three further limits are narrower. CAST holds a memory of $1{,}200$ utterances and may weaken when conditions change faster than useful neighborhoods form; its margin and temperature are in raw score units and were carried to WavLM unchanged; and CAST-LN does not retain source performance on WavLM without labeled replay. The streams have their own limits: GENERATOR excludes the highest-error frozen regime and runs one fixed generator order, LOCALE varies locale and recording condition together, and CAST was developed on one XLS-R checkpoint of the same GENERATOR construction used for evaluation rather than on a disjoint tuning stream (Appendix C.3).

Two things are now established for this setting. Discrimination and the operating point fail separately, so evaluation at one threshold cannot tell the two apart, and a method that lowers delivered cost has not thereby repaired the detector. And the ordering information that prediction-driven objectives lack is available elsewhere: relations between utterances in the frozen representation carry it, without labels, thresholds, or a class prior. CAST does not find the operating point, because its loss reads only score differences, and label-free recalibration recovers only part of the calibration loss because the shift is not pure label shift. The immediate test for any successor is a target stream with many bonafide speakers, which would show whether relational signals of this kind survive without repeated-speaker structure.

AI USE STATEMENT

The research question, the benchmark, the CAST method, and the analysis in this paper are the authors' own work. On the required disclosure list, generative AI tools were used for one task: code implementing the methods and the data preparation was written by an agentic coding assistant from task specifications written by the authors. We did not use these tools to develop the conceptual framework, to formulate or refine hypotheses, to formulate mathematical claims, to supply ingredients for proofs, or to write proofs. Generating synthetic datasets, translation, transcribing recordings, and qualitative or thematic analysis do not apply to this work.

On the recommended list, we used these tools to write and edit software code, and to edit the manuscript for clarity and length.

We have reviewed all AI-assisted work. Every number in the paper and its appendices was traced back to a stored run. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Section 3 defines the protocol, the five streams, the cost function and its decomposition, and the corpus controls applied at dataset construction; Appendices A.1, A.3 and A.6 give the full stream definitions, the generator inclusion rule and ordering, and the audit that follows the controls. Section 5 states CAST's hyperparameters, adapted parameter sets, and selection procedure, with the remaining implementation details in Appendix C. Baseline configurations and reference arms are in Appendix B, and Appendix A.5 gives the aggregation and bootstrap procedure behind every reported interval. Per-checkpoint results for all streams are in Appendix F.

REFERENCES

Amr Alexandari, Anshul Kundaje, and Avanti Shrikumar. Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In *International Conference on Machine Learning (ICML)*, 2020.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. XLS-R: Self-supervised cross-lingual speech representation learning at scale. In *Proc. Interspeech*, 2022.

Maximilian Bisani and Hermann Ney. Bootstrap estimates for confidence intervals in ASR performance evaluation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2004.

Niko Brümmer and Johan du Preez. Application-independent evaluation of speaker detection. *Computer Speech & Language*, 20(2–3):230–275, 2006.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *International Conference on Machine Learning (ICML)*, 2005.

Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022a.

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, 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, 2022b.

Xiaohuan Chen, Wenhuan Lu, Ruiteng Zhang, Junhai Xu, Xugang Lu, Lin Zhang, and Jianguo Wei. Continual unsupervised domain adaptation for audio deepfake detection. In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025a. doi: 10.1109/ICASSP49660.2025.10890538.

Yujie Chen, Jiangyan Yi, Cunhang Fan, Jianhua Tao, Yong Ren, Siding Zeng, Chu Yuan Zhang, Xinrui Yan, Hao Gu, Jun Xue, Chenglong Wang, Zhao Lv, and Xiaohui Zhang. Region-based optimization in continual learning for audio deepfake detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23651–23659, 2025b. doi: 10.1609/aaai.v39i22.34535.

Ziyang Chen, Yiwen Ye, Yongsheng Pan, and Yong Xia. Gradient alignment improves test-time adaptation for medical image segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 2429–2437, 2025c. doi: 10.1609/aaai.v39i3.32244.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1993.

Yassine El Kheir, Younes Samih, Suraj Maharjan, Tim Polzehl, and Sebastian Möller. Comprehensive layer-wise analysis of SSL models for audio deepfake detection. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4070–4082. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.227.

João Gama, Raquel Sebastião, and Pedro Pereira Rodrigues. On evaluating stream learning algorithms. *Machine Learning*, 90(3):317–346, 2013.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Sachin Goyal, Mingjie Sun, Aditi Raghunathan, and J. Zico Kolter. Test time adaptation via conjugate pseudo-labels. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, 2017.

Minguk Jang, Sae-Young Chung, and Hye Won Chung. Test-time adaptation via self-training with nearest neighbor information. In *International Conference on Learning Representations (ICLR)*, 2023.

Kartik Jhawar and Lipo Wang. When is test-time adaptation identifiable from unlabeled evidence? arXiv preprint arXiv:2609.11235, 2026.

Jee-weon Jung, Hee-Soo Heo, Hemlata Tak, Hye-jin Shim, Joon Son Chung, Bong-Jin Lee, Ha-Jin Yu, and Nicholas Evans. AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Tomi Kinnunen, Kong Aik Lee, Héctor Delgado, Nicholas Evans, Massimiliano Todisco, Md Sahidullah, Junichi Yamagishi, and Douglas A. Reynolds. t-DCF: A detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification. In *Proc. The Speaker and Language Recognition Workshop (Odyssey)*, 2018.

Ivan Kukanov, Janne Laakkonen, Tomi Kinnunen, and Ville Hautamäki. Meta-learning approaches for improving detection of unseen speech deepfakes. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pp. 1173–1178. IEEE, 2024. doi: 10.1109/SLT61566.2024.10832350.

Janne Laakkonen, Ivan Kukanov, and Ville Hautamäki. Generalizable speech deepfake detection via meta-learned LoRA. In *2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 19092–19096. IEEE, 2026. doi: 10.1109/ICASSP55912.2026.11463170.

Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on Challenges in Representation Learning*, 2013.

Jonghyun Lee, Dahuin Jung, Saehyung Lee, Junsung Park, Juhyeon Shin, Uiwon Hwang, and Sungroh Yoon. Entropy is not enough for test-time adaptation: From the perspective of disentangled factors. In *International Conference on Learning Representations (ICLR)*, 2024.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, 2020.

Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025.

Zachary C. Lipton, Yu-Xiang Wang, and Alexander J. Smola. Detecting and correcting for label shift with black box predictors. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3122–3130. PMLR, 2018.

Haoxin Ma, Jiangyan Yi, Jianhua Tao, Ye Bai, Zhengkun Tian, and Chenglong Wang. Continual learning for fake audio detection. In *Interspeech 2021*, 2021.

Nicolas M. Müller, Franziska Dieckmann, Pavel Czempin, Roman Canals, Konstantin Böttinger, and Jennifer Williams. Speech is silver, silence is golden: What do ASVspoof-trained models really learn? In *2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pp. 55–60, 2021. doi: 10.21437/ASVSPOOF.2021-9.

Nicolas M. Müller, Pavel Czempin, Franziska Dieckmann, Adam Froghyar, and Konstantin Böttinger. Does audio deepfake detection generalize? In *Interspeech 2022*, 2022.

Nicolas M. Müller, Nicholas Evans, Hemlata Tak, Philip Sperl, and Konstantin Böttinger. Harder or different? understanding generalization of audio deepfake detection. In *Proc. Interspeech*, 2024a.

Nicolas M. Müller, Piotr Kawa, Wei Herng Choong, Edresson Casanova, Eren Gölge, Thorsten Müller, Piotr Syga, Philip Sperl, and Konstantin Böttinger. MLAAD: The multi-language audio anti-spoofing dataset. In *Proc. International Joint Conference on Neural Networks (IJCNN)*, 2024b.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16888–16905. PMLR, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations (ICLR)*, 2023.

Hideyuki Oiso, Yuto Matsunaga, Kazuya Kakizaki, and Taiki Miyagawa. Prompt tuning for audio deepfake detection: Computationally efficient test-time domain adaptation with limited target dataset. In *Interspeech 2024*, pp. 2710–2714, 2024. doi: 10.21437/Interspeech.2024-81.

Ori Press, Steffen Schneider, Matthias Kümmerer, and Matthias Bethge. RDumb: A simple approach that questions our progress in continual test-time adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023. doi: 10.52202/075280-1735.

Danfeng Qin, Stephan Gammeter, Lukas Bossard, Till Quack, and Luc Van Gool. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11:2487–2531, 2010.

Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 14(1):21–41, 2002.

Sabyasachi Sahoo, Mostafa ElAraby, Jonas Ngnawe, Yann Batiste Pequignot, Frédéric Precioso, and Christian Gagné. A layer selection approach to test time adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20237–20245, 2025. doi: 10.1609/aaai.v39i19.34229.

Pierre Serrano, Raphaël Duroselle, Florian Angulo, Jean-François Bonastre, and Olivier Boeffard. Improving out-of-domain audio deepfake detection via layer selection and fusion of SSL-based countermeasures. arXiv preprint arXiv:2509.12003, 2025.

Imdat Solak. The M-AILABS speech dataset, 2019. `https://www.caito.de/2019/01/the-m-ailabs-speech-dataset/`.

Hemlata Tak, Massimiliano Todisco, Xin Wang, Jee-weon Jung, Junichi Yamagishi, and Nicholas Evans. Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation. In *The Speaker and Language Recognition Workshop (Odyssey 2022)*, pp. 112–119, 2022. doi: 10.21437/Odyssey.2022-16.

Jiaxi Tang and Ke Wang. Ranking distillation: Learning compact ranking models with high performance for recommender system. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018.

Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee. ASVspoof 2019: Future horizons in spoofed and fake audio detection. In *Proc. Interspeech*, pp. 1008–1012, 2019.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 2021.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Xin Wang, Junichi Yamagishi, Massimiliano Todisco, Héctor Delgado, Andreas Nautsch, Nicholas Evans, Md Sahidullah, Ville Vestman, Tomi Kinnunen, Kong Aik Lee, Lauri Juvela, Paavo Alku, Yu-Huai Peng, Hsin-Te Hwang, Yu Tsao, Hsin-Min Wang, Sébastien Le Maguer, Markus Becker, Fergus Henderson, Rob Clark, Yu Zhang, Quan Wang, Ye Jia, Kai Onuma, Koji Mushika, Takashi Kaneda, Yuan Jiang, Li-Juan Liu, Yi-Chiao Wu, Wen-Chin Huang, Tomoki Toda, Kou Tanaka, Hirokazu Kameoka, Ingmar Steiner, Driss Matrouf, Jean-François Bonastre, Avashna Govender, Srikanth Ronanki, Jing-Xuan Zhang, and Zhen-Hua Ling. ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech. *Computer Speech & Language*, 64: 101114, 2020.

Xin Wang, Héctor Delgado, Hemlata Tak, Jee-weon Jung, Hye-jin Shim, Massimiliano Todisco, Ivan Kukanov, Xuechen Liu, Md Sahidullah, Tomi Kinnunen, Nicholas Evans, Kong Aik Lee, and Junichi Yamagishi. ASVspoof 5: Crowdsourced speech data, deepfakes, and adversarial attacks at scale. In *The Automatic Speaker Verification Spoofing Countermeasures Workshop (ASVspoof 2024)*, pp. 1–8, 2024. doi: 10.21437/ASVspoof.2024-1.

Xin Wang, Wanying Ge, and Junichi Yamagishi. Towards data drift monitoring for speech deepfake detection in the context of MLOps. In *2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2026. doi: 10.1109/ICASSP55912.2026.11464546.

Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, and Héctor Delgado. ASVspoof 2021: Accelerating progress in spoofed and deepfake speech detection. In *Proc. ASVspoof Challenge Workshop*, 2021.

Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15922–15932, 2023. doi: 10.1109/CVPR52729.2023.01528.

Xiaohui Zhang, Jiangyan Yi, Chenglong Wang, Chu Yuan Zhang, Siding Zeng, and Jianhua Tao. What to remember: Self-adaptive continual learning for audio deepfake detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19569–19577, 2024. doi: 10.1609/aaai.v38i17.29929.

Hao Zhao, Yuejiang Liu, Alexandre Alahi, and Tao Lin. On pitfalls of test-time adaptation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 42058–42080. PMLR, 2023.

Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Zhi Zhou, Ming Yang, Shi-Yu Tian, Kun-Yang Yu, Lan-Zhe Guo, and Yu-Feng Li. On the learnability of test-time adaptation: A recovery complexity perspective. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

A Benchmark Construction and Validity

A.1 Deployment protocol

We distinguish four units. A *stream* is a fixed-order sequence of unlabeled utterances. A *batch* contains the $B = 16$ utterances presented at one step. A *block* is a contiguous part of a stream in which the designated varying factor is fixed. A *window* contains 600 consecutive utterances and is the reporting unit for single-block streams. We use R for the spoofed fraction.

Evaluation is strictly prequential. Each batch is scored and its scores are recorded before any update using that batch. Recorded scores are never revised, so every utterance contributes exactly one deployment-time score. A method may retain information from earlier utterances but cannot access future ones. Methods receive no target labels, domain or block identifiers, block boundaries, externally supplied stream position or length, or source data. Every stream starts independently from the same source checkpoint, and the benchmark imposes no reset at block boundaries. A method may reset itself using information available online, as RDumb does (Press et al., 2023).

Each utterance is scored from the first $64{,}600$ samples ($4.04\,\mathrm{s}$) remaining after a 50 ms trim at each edge. The trim is applied identically to the frozen detector and every adaptive method, so it introduces no input difference between methods. Its role in controlling edge-silence shortcuts is described in Appendix A.6.

A.2 Stream construction

Source. We use the ASVspoof 2019 LA evaluation partition at its natural class prior, with each utterance appearing at most once. It retains the source crop and wrap-padding rule rather than the target minimum-length rule, while receiving the same scoring-time controls as the target streams. Source measures retention in the source regime. It is not strictly shift-free, because most evaluation attacks were unseen during source training.

Stationary. The target distribution is fixed at $R = 0.5$. Spoofed utterances are drawn round-robin from the 52 admitted generators so that each 600-utterance window contains approximately the same generator mixture. This stream measures continual adaptation after a fixed target shift.

Generator. The stream contains 52 blocks of $1{,}200$ utterances. Each block contains one synthesis system and 600 bonafide utterances, giving $R = 0.5$. Blocks are ordered by increasing frozen difficulty. Because difficulty therefore changes together with adaptation history, we use difficulty terciles only for descriptive summaries and not as causal interventions.

Prior. The stream contains five blocks of $12{,}000$ utterances with spoofed fractions

$$R = 0.50 \rightarrow 0.20 \rightarrow 0.05 \rightarrow 0.20 \rightarrow 0.50.$$

These fractions are enforced within each 600-utterance window rather than within individual batches. The repeated $R = 0.50$ condition provides a retention diagnostic: blocks 0 and 4 have the same class prior, while FAR and FRR are class-conditional. A systematic difference between the two blocks therefore cannot be attributed to class prior alone.

LOCALE. The stream contains eight blocks of $1{,}200$ utterances at $R = 0.5$. The generator family is fixed while the bonafide locale varies across eight languages, seven of them non-English. Locale and recording condition are confounded in M-AILABS, so this stream serves only as a breadth check. Differences between blocks cannot be attributed specifically to language, accent, geography, or recording environment.

A.3 GENERATOR INCLUSION AND ORDERING

GENERATOR uses 52 of the 116 English generators in MLAAD v10, released on 18 May 2026. This release contains 175 text-to-speech models across 54 languages (Müller et al., 2024b). Because MLAAD is released as a growing series and its generator inventory changes between versions, all selection results here refer specifically to v10.

A generator is included when its min-HTER under the development XLS-R checkpoint lies in $[5\%, 25\%)$. We fixed the inclusion rule, resulting generator set, and frozen-difficulty ordering before the final evaluation. The same generators and ordering are used for every checkpoint and both backbones.

This rule excludes the highest-error frozen regime and therefore limits what GENERATOR establishes. The exact ordering of individual generators also does not transfer perfectly across checkpoints or backbones. The coarser structure used for reporting does transfer: the three difficulty terciles remain consistently ordered. We therefore use the terciles descriptively and do not treat the position of an individual generator as an independently controlled difficulty intervention.

A.4 THRESHOLD DEFINITION AND ROBUSTNESS

For each source checkpoint, the deployed threshold θ is the EER threshold estimated on the ASVspoof 2019 LA development partition. It was fixed independently of the target DCF results and is never retuned using target labels.

The development EER threshold and development minDCF threshold optimize different source-domain criteria and need not coincide. We therefore repeat the evaluation using the source-development minDCF threshold for each checkpoint. The conclusions concerning prediction-driven adaptation, recalibration, labeled updates, and CAST are unchanged.

Threshold estimation and stream evaluation use the same scoring path, including the 50 ms edge trim of Appendix A.1. Recomputing θ from the stored development scores reproduces the deployed value for every checkpoint. The development partition shares no utterance keys with any evaluation stream.

A.5 AGGREGATION AND UNCERTAINTY

An adapting detector can change its score scale as a stream progresses. Sweeping one threshold over a long pooled stream can therefore mix regions with different score scales. The resulting metric can reflect both within-region discrimination and score movement across time. We therefore compute threshold-optimized metrics locally.

On GENERATOR and PRIOR, each block receives its own threshold sweep, and summaries are means over the relevant blocks. GENERATOR is additionally summarized by difficulty tercile and PRIOR by block position. The single-block SOURCE and STATIONARY streams use consecutive 600-utterance windows, giving 118 and 106 windows, respectively.

We check whether these reporting units determine the conclusions. A SOURCE window contains about 60 bonafide trials, giving an FRR resolution of approximately 1.7 percentage points. Grouping

three consecutive windows into 1,800-utterance units does not change any conclusion. We also repeat the minDCF analysis on shorter segments within the longer blocks. No conclusion for CAST or either labeled reference changes. On PRIOR, the only affected baseline cells are nine effects whose primary estimates lie within ± 0.005 of zero.

Scores and trials are never pooled across checkpoints because each checkpoint has its own score scale and frozen reference. Every inferential comparison is a within-checkpoint paired difference between a method and the frozen detector on the same utterances. Bootstrap intervals are stratified by class and by block, or by window for the two single-block streams (Efron & Tibshirani, 1993; Bisani & Ney, 2004). Means reported across checkpoints are means of the separate checkpoint estimates rather than estimates obtained by pooling scores.

A.6 CORPUS-SHORTCUT AUDIT

Target bonafide and spoofed speech come from different corpora and therefore contain non-semantic properties correlated with class. We audit four such cue families: native sample rate, non-speech at utterance boundaries, signal level, and duration or padding behavior. Each candidate cue is evaluated as a standalone classifier of the target class. Lower EER indicates a stronger shortcut, while 50% indicates chance-level discrimination. These tests cover cues specified in advance; they do not bound all information available in the learned representation, which is examined separately in Appendix D.2.

Sample rate. Native sample rate identifies the two corpora perfectly from metadata: MLAAD is stored at $22{,}050$ Hz (Müller et al., 2024b), whereas M-AILABS is stored at 16 kHz. Resampling alone does not remove all evidence of the original rate. Residual anti-aliasing structure in the decoded waveform separates the corpora at 0.47% EER. We therefore resample with a low-pass filter applied symmetrically to both target classes and to the ASVspoof 2019 LA training data.

Edge silence. Leading and trailing non-speech is a documented shortcut in ASVspoof detectors (Müller et al., 2021). After removing exact-zero samples, low-level edge non-speech still separates the target corpora at 12.2% EER. Its class orientation also reverses between ASVspoof 2019 and the target corpora. A source-trained association with this cue can therefore become misleading after the shift. We remove exact-zero edge samples in preprocessing and apply the 50 ms scoring-time trim described above, following the trimming protocol used for the ASVspoof 2021 hidden track (Yamagishi et al., 2021).

Signal level. We apply per-utterance peak normalization to both target classes and to the source training data. After all controls, residual level structure is the strongest remaining audited cue, separating the target classes at 42.7–42.9% EER. No preprocessing step uses statistics computed across utterances, batches, or streams.

Duration and wrap padding. Short source-training utterances are wrap-padded to the scoring crop. On the source training partition, 99.1% of bonafide and 89.9% of spoofed utterances are shorter than this crop. The resulting repetition cue is weak as a standalone classifier (45% EER), and the frozen detector shows no clear evidence of using it. The target streams further suppress this cue by including only utterances at least as long as the scoring crop before edge trimming. After stream construction and scoring-time preprocessing, only 1.5–2.1% of utterances in each class require padding, with a bona-fide-to-spoof padding ratio between 1.01 and 1.06.

Effect of the controls. The controls make source evaluation substantially harder. Under the same training recipe, the three XLS-R checkpoints obtain ASVspoof 2019 LA evaluation EERs of 0.44%, 0.68%, and 1.41% on uncontrolled audio, compared with 5.33%, 6.23%, and 5.97% on the controlled audio used by the benchmark. These checks do not establish that the paired target corpora are free of all corpus-specific information. They show that the strongest measured shortcuts are removed or substantially weakened before adaptation is evaluated.

B EXPERIMENTAL SETUP AND BASELINE CHECKS

B.1 SOURCE MODELS AND CHECKPOINT SELECTION

We train three source models for each backbone, XLS-R and WavLM Large, with an AASIST back-end on the controlled ASVspoof 2019 LA training partition. Training uses a fixed budget of 30 epochs. For each run, we retain the checkpoint with the lowest development-set negative log-likelihood. This selection rule was fixed before any target stream was scored and applied identically to all runs.

We do not select checkpoints by development EER because the source models nearly saturate the development partition. With only $2{,}548$ utterances, a difference of one or two trials can determine the EER ordering between otherwise similar checkpoints. Negative log-likelihood therefore provides a finer checkpoint-selection criterion in this regime.

B.2 ADAPTED PARAMETERS AND REFERENCE ARMS

Table 3 lists the parameter sets updated by each adapting arm. Tent, EATA-rd, RDumb, and RoTTA update $110{,}594$ normalization affine parameters: the transformer LayerNorm affines and the AASIST BatchNorm affines. SAR and CAST-LN update only the $108{,}544$ transformer LayerNorm affines. Appendix C.4 directly tests whether this small difference in parameter sets explains the comparison between Tent and CAST.

Labeled-LN updates the same $110{,}594$ parameters as Tent. It uses the same optimizer, learning rate, batch size, and prequential update order, but replaces the label-free objective with cross-entropy against the true target labels. Labeled-LN is therefore a non-deployable reference for the discrimination improvement available when correct labels are supplied to the same parameter set.

Labeled-FT is the higher-capacity labeled reference on the target streams. It updates the same $311{,}227{,}520$ transformer-trunk parameters as CAST-FT, with both the convolutional feature encoder and the AASIST back-end frozen, and replaces the label-free objective with cross-entropy against the true target labels. A separate labeled diagnostic was run on SOURCE with the trunk and AASIST back-end both updated. That diagnostic contains $311{,}674{,}762$ trainable parameters and is reported separately from Labeled-FT.

The frozen detector with MLLS recalibration (Saerens et al., 2002) provides a different reference. MLLS estimates the target class proportion from recent unlabeled scores and adjusts the operating threshold without updating detector parameters or changing score ordering. Its recovery of calibration loss is reported in Appendix F.

Table 3: Parameter sets updated by each adapting arm. Counts are taken directly from the corresponding optimizer parameter groups. The labeled SOURCE diagnostic is separate from the target-stream Labeled-FT reference.

Arm	Parameters	Updated parameters
Tent, EATA-rd, RDumb, RoTTA	110,594	transformer LayerNorm + AASIST BatchNorm affines
SAR, CAST-LN	108,544	transformer LayerNorm affines
Labeled-LN	110,594	transformer LayerNorm + AASIST BatchNorm affines
CAST-FT	311,227,520	transformer trunk outside feature encoder
Labeled-FT	311,227,520	transformer trunk outside feature encoder
Labeled SOURCE diagnostic	311,674,762	transformer trunk outside feature encoder + AASIST

All five prediction-driven baselines, Labeled-LN, Labeled-FT, and CAST-FT are evaluated with XLS-R. SAR, RoTTA, and CAST-LN are additionally evaluated with WavLM. The SOURCE stream itself is evaluated only with XLS-R; the wider labeled SOURCE diagnostic is used only for the operating-point analysis.

B.3 Checks on weak baseline movement

SAR and EATA-rd require additional checks because their discrimination effects are close to zero. We test whether these results can be explained by negligible parameter updates rather than by the adaptation objectives themselves.

SAR. SAR changes minDCF by less than 0.001 in magnitude throughout the evaluated streams. Its sample filter admits nearly all samples, and its gradients remain finite. The near-zero result is therefore not explained by rejected samples or non-finite gradients. With the published SGD optimizer and learning rate, its relative parameter displacement is approximately 2×10^{-8} per step. Increasing the learning rate to as much as 2.5×10^{-4} raises this displacement to approximately 5×10^{-6} per step but does not produce a meaningful discrimination improvement. We therefore describe SAR as effectively inert under the settings tested here. This result does not establish that selective normalization adaptation is ineffective in general.

EATA-rd. EATA-rd also uses SGD and updates its parameters, but its displacement at the matched setting is approximately one tenth of Tent's. We therefore evaluate four learning rates spanning three orders of magnitude, including the published setting. At the larger rates, EATA-rd reaches parameter displacement comparable to Tent's, but its effect on local minDCF remains negligible or harmful. Its weak discrimination result is therefore not explained simply by insufficient parameter movement.

B.4 Tent tuning control

To test whether the comparison with CAST is explained by giving Tent less opportunity for configuration selection, we evaluate 16 Tent configurations on the same XLS-R GENERATOR development checkpoint used for CAST. The search varies learning rate, adapted parameter set, number of update steps, and logit temperature, and uses the same offline selection statistic as the CAST development step.

The selected configuration adapts the transformer LayerNorm parameters at learning rate 10^{-5}. It improves local minDCF by 0.0364 on checkpoint 1234, on which it was selected, but does not transfer consistently to the two held-out checkpoints: its changes are $+0.0706$ on checkpoint 1235 and -0.0272 on checkpoint 1236. The corresponding published Tent configuration changes minDCF by -0.0127 and -0.0101, while CAST-LN changes it by -0.0617 and -0.0674.

Table 4: Tent tuning control on GENERATOR. Checkpoint 1234 is the development checkpoint used for configuration selection; checkpoints 1235 and 1236 are held out. Entries are changes in local minDCF relative to the frozen detector.

Method	1234	1235	1236	Held-out mean
Selected Tent	$-.0364$	$+.0706$	$-.0272$	$+.0217$
Published Tent	$-.0034$	$-.0127$	$-.0101$	$-.0114$
CAST-LN	$-.0670$	$-.0617$	$-.0674$	$-.0645$

Selection on checkpoint 1234 therefore does not transfer consistently to the held-out checkpoints. The result does not support explaining the CAST–Tent discrimination gap by the absence of a Tent tuning stage.

C CAST Implementation and Development

C.1 Selected configuration

Table 5 summarizes the CAST-LN configuration used in the reported experiments. The frozen embedding h_i is the time-averaged output of transformer encoder layer 4 (zero-indexed, corresponding to the fifth transformer block). The adapting detector updates only the transformer LayerNorm affine parameters using Adam (Kingma & Ba, 2015). Optimizer state is carried across batches and is not reset during a stream.

The representation layer was selected during development rather than fixed from prior work. Its location is nevertheless consistent with previous layer-wise analyses of self-supervised speech mod-

Table 5: CAST-LN configuration fixed after development.

Component	Selected setting
Representation	transformer layer 4, mean-pooled over time
Similarity	cosine after pool centering and ℓ_2 normalization
Neighborhood	$k = 10$, reciprocal neighbors only
Memory	$N = 1{,}200$ utterances, FIFO
Pair margin	$m = 1$
RankNet temperature	$T = 2$
Optimizer	Adam
Learning rate	10^{-5}
Updated parameters	$108{,}544$ transformer LayerNorm affines

els, which identify intermediate transformer layers as carrying useful transferable information for this task (Serrano et al., 2025; El Kheir et al., 2025).

C.2 Reciprocal neighborhoods

We use reciprocal rather than ordinary k-nearest-neighbor sets. On the development configuration, ordinary neighborhoods degraded pseudo-score ordering under severe class imbalance, whereas reciprocal neighborhoods improved it. Reciprocity was therefore retained as part of the selected neighborhood signal.

This behavior is consistent with hubness in high-dimensional nearest-neighbor spaces (Radovanović et al., 2010), where some points occur in unusually many neighbor lists. Under class imbalance, such points can increase majority-class contamination of minority-class neighborhoods. Requiring both $j \in \mathcal{N}_k(i)$ and $i \in \mathcal{N}_k(j)$ retains only mutual neighbor relations and removes part of this contamination. Reciprocity does not eliminate the problem: residual contamination in the 5%-spoof block of Prior is reported in Appendix D.3.

C.3 Development protocol

The adapted parameter set and optimizer were fixed before hyperparameter development. We then selected the representation layer, neighborhood size, memory length, reciprocity, margin, temperature, ranking objective, pair-admission rule, and learning rate on checkpoint 1234 of Generator. The search space and selection procedure were fixed before the sweep. Target labels were used only to evaluate candidate configurations offline; CAST receives no target labels during online adaptation.

After development, the configuration in Table 5 was fixed for all remaining experiments. Checkpoints 1235 and 1236 of Generator were not used for development. Neither were Stationary, Prior, Locale, or any WavLM experiments. For this reason, the main paper reports CAST results on Generator separately for the two held-out checkpoints.

A development stream disjoint from all evaluation data would provide stronger separation. The available corpus pairing does not permit such a construction. M-AILABS English contains only four bonafide speakers, and all four occur in the evaluation streams. Holding out generators would therefore leave the bonafide side shared, while holding out speakers would materially change the target distribution. We instead rely on held-out source checkpoints and stream constructions not used during development, and treat the absence of a fully disjoint development corpus as a limitation.

C.4 Parameter-set controls

CAST-LN updates $108{,}544$ transformer LayerNorm affine parameters. Tent, EATA-rd, RDumb, RoTTA, and Labeled-LN additionally update the $2{,}050$ AASIST BatchNorm affine parameters, for $110{,}594$ parameters in total. We test whether this difference accounts for the comparison with Tent by defining CAST-Norm, which applies the CAST objective to exactly the $110{,}594$ parameters updated by Tent.

At the selected CAST learning rate of 10^{-5}, CAST-Norm changes held-out GENERATOR minDCF by -0.0650, compared with -0.0646 for CAST-LN. The corresponding EERs are 12.35% and 12.34%. Adding the AASIST BatchNorm affine parameters therefore has negligible effect in this comparison and does not account for the reported CAST gain.

We next match both the parameter set and learning rate to Tent. At Tent's learning rate of 10^{-6}, CAST-Norm changes local minDCF by -0.0212, compared with -0.0114 for Tent. The difference between the objectives therefore remains when both adapted parameters and learning rate are matched. Increasing the CAST learning rate to the selected 10^{-5} setting increases the magnitude of the gain, but is not necessary for the discrimination improvement to appear.

C.5 CAST-FT

CAST-FT uses the same neighborhood signal and ranking objective as CAST-LN but increases the number of adapted parameters. It updates all $311{,}227{,}520$ transformer-trunk parameters outside the convolutional feature encoder at a learning rate of 10^{-6}. The convolutional feature encoder and AASIST back-end remain frozen.

CAST-FT was first evaluated on the development configuration as a test of whether the CAST gain was specific to LayerNorm adaptation. After the positive development result, we extended it to the full XLS-R benchmark. Its broader evaluation should therefore be interpreted as an exploratory analysis of adaptation capacity rather than as an independently developed configuration. CAST-FT was not evaluated with WavLM, so all cross-backbone evidence for CAST comes from CAST-LN.

D NEIGHBORHOOD SIGNAL AND ITS LIMITATIONS

D.1 DIRECT SIGNAL SCREEN

Before using a candidate signal for adaptation, we evaluate it directly as a scorer. No parameters are updated. Each utterance receives the candidate pseudo-score, and its local minDCF is compared with that of the frozen detector. This screen asks whether the candidate contains information that can improve score ordering. It does not test whether the resulting scorer is suitable for deployment; that question is addressed separately in Appendix E.1.

Pseudo-scores derived only from an utterance's own scalar score do not improve the ordering. In contrast, averaging the frozen score over reciprocal nearest neighbors in the frozen representation produces substantial improvements. The signal was developed on XLS-R checkpoint 1234 of GENERATOR. On the two checkpoints not used for development, minDCF changes by -0.095 and -0.104 on GENERATOR, -0.067 and -0.096 on STATIONARY, and -0.076 and -0.106 on PRIOR.

The reciprocal neighborhoods are also strongly label-pure. On GENERATOR, approximately 95% of reciprocal neighbors share the query label on both backbones, compared with about 50% after labels are permuted within blocks. Purity remains about 91% among utterances that the frozen detector misclassifies at the deployed threshold θ. These values vary by less than half a percentage point across the three checkpoints. The neighborhood relation therefore contains class-relevant information beyond the frozen detector's binary decision.

D.2 WHAT THE NEIGHBORHOODS ENCODE

High label purity does not establish what produces the neighborhood structure. Bonafide and spoofed target speech come from different corpora, and intermediate self-supervised representations can encode corpus, speaker, channel, and synthesis information. We therefore probe the frozen layer-4 representation directly.

The probes use the same centered and ℓ_2-normalized embeddings as CAST. We evaluate linear and k-nearest-neighbor probes with $k = 10$ on GENERATOR and STATIONARY for XLS-R and on GENERATOR for WavLM, using all three checkpoints. M-AILABS utterances are grouped by book, with each book corresponding to a recording session. We therefore also report a grouped split in which complete books are held out. Chance levels are computed from the realized target proportions.

Three observations matter.

Table 6: Accuracy of probes on frozen layer-4 embeddings. "Grouped" holds out complete M-AILABS books, so recording sessions do not appear on both sides of a split. The source speaker of a spoofed utterance is the M-AILABS voice on which its synthesis system was trained.

Target	Probe	Ungrouped	Grouped	Chance
Corpus, XLS-R	linear	0.997	0.980	0.50
Corpus, XLS-R	k-NN	0.965	0.831	0.50
Corpus, WavLM	linear	0.995	0.978	0.50
Corpus, WavLM	k-NN	0.957	0.869	0.50
Speaker, bonafide	linear	1.000	0.998	0.345
Speaker, bonafide	k-NN	1.000	0.997	0.345
Source speaker, spoofed	linear	0.40	0.30	0.341
Source speaker, spoofed	k-NN	0.33	0.30	0.341
Generator, spoofed	linear	0.916	–	0.019
Generator, spoofed	k-NN	0.823	0.823	0.019

First, corpus identity remains highly recoverable. Under the grouped split, linear accuracy is 0.980 for XLS-R and 0.978 for WavLM. Nearest-neighbor corpus accuracy falls more substantially, from 0.965 to 0.831 on XLS-R and from 0.957 to 0.869 on WavLM. The cross-corpus pairing therefore leaves substantial corpus information in the representation even after the waveform-level controls of Appendix A.6.

We directly test how much of the neighborhood-smoothing gain can be attributed to linearly decodable corpus information. We project out two corpus-predictive directions fitted on GENERATOR and recompute the direct neighborhood scorer. Across the three XLS-R checkpoints, the projected representation retains 84% of the original discrimination gain on GENERATOR, 45% on STATIONARY, and 52% on PRIOR. Removing a matched two-dimensional random subspace does not produce the same loss. After corpus-direction removal, linear corpus prediction falls close to chance, while k-NN corpus accuracy remains around 0.78. The GENERATOR gain is therefore not reducible to the fitted linear corpus directions, whereas the gains on STATIONARY and PRIOR depend substantially on them. Because substantial local corpus information remains after projection, this control does not establish corpus independence.

Second, the bonafide side is strongly organized by speaker. Under the session-disjoint split, speaker identity is recovered at 0.997 by nearest-neighbor retrieval. More directly, on GENERATOR, a bonafide utterance's reciprocal neighbors share its speaker 99.9% of the time, compared with 26.4% for a random draw matched to the causal pool available at that point in the stream. The four recurring M-AILABS speakers therefore provide strong relational structure on the bonafide side.

Third, the spoofed side has a different structure. The source speaker on which a synthesis system was trained is recovered at or below chance, whereas generator identity reaches 0.823 by nearest-neighbor retrieval and 0.916 with a linear probe, against a 1.9% chance baseline. The spoofed representation is therefore organized strongly by generator rather than by source speaker.

These measurements establish what information is available in the neighborhoods, not which factor causally produces CAST's discrimination gain. In particular, high speaker recoverability does not establish that CAST depends on speaker identity. The corpus-direction control likewise removes only linearly decodable corpus information and leaves substantial local corpus structure. The benchmark cannot cleanly separate these dependencies because its bonafide side contains only four recurring English speakers and corpus identity remains correlated with class. A target stream containing bonafide and spoofed speech under a same-corpus construction, with many bonafide speakers and little or no recurrence, would be needed to test these dependencies more directly.

D.3 FAILURE UNDER SEVERE CLASS IMBALANCE

The weakest CAST-LN result occurs in the 5%-spoof block of PRIOR. Relative to the frozen detector, local minDCF changes by $+0.035$, $+0.052$, and $+0.004$ across the three XLS-R checkpoints.

The reciprocal-neighbor structure also deteriorates in this block. Approximately 24% of spoofed utterances have no reciprocal neighbor, compared with about 4% under the balanced prior. Among

spoofed utterances that retain reciprocal neighbors, 12.6% of those neighbors are bonafide, compared with approximately 5% under the balanced prior.

The coincidence between weaker neighborhoods and weaker CAST-LN performance is consistent with class imbalance reducing the quality of the available relational signal. It does not establish that neighborhood contamination causes the performance loss: CAST-FT improves discrimination in the same 5%-spoof block while using the same neighborhoods. We therefore treat these measurements as evidence of a limitation in the signal under severe imbalance, not as a causal explanation of the CAST-LN result.

E Alternative Uses of the Signal and Operating-Point Controls

E.1 Direct neighborhood smoothing

The neighborhood-smoothed pseudo-score of equation 2 can be used directly as the detector score without updating any parameters. This scorer remains causal and label-free: each score uses only the current batch and the memory available at that point in the stream. Its discrimination therefore measures how much useful ordering information is already present in the neighborhood signal before that signal is distilled into an adapting model.

On XLS-R checkpoints 1235 and 1236, the direct scorer changes local minDCF by -0.082 on Stationary, -0.100 on Generator, -0.091 on Prior, and -0.033 on Locale. On the same two checkpoints, the corresponding changes for CAST-LN are -0.069, -0.065, -0.023, and -0.012. Direct smoothing therefore improves target ordering more than CAST-LN on each target stream under the same checkpoint basis.

The stronger discrimination does not make the direct scorer a complete deployment solution. After the smoothed scorer and frozen detector are independently recalibrated using causally estimated label-free thresholds, the smoothed scorer delivers lower cost on only one of the three XLS-R checkpoints on each main target stream and on none of the three Locale checkpoints. On WavLM, it delivers lower cost on two of three checkpoints for Stationary and Prior, and on all three for Generator.

A second limitation appears on the source-regime control. Direct smoothing worsens local minDCF on 117 of 118 Source windows and increases its mean by 0.112; EER rises from 6.3% to 12.4%. In contrast, CAST-LN lowers source minDCF by 0.017. The same neighborhood operation that improves ordering after target shift can therefore damage an ordering that remains suitable. The direct smoothed scorer was not evaluated on WavLM Source.

E.2 Class targets from the same neighborhoods

Prior neighborhood-based TTA methods commonly use nearby samples to construct class targets or contrastive objectives (Jang et al., 2023; Yang et al., 2021; Chen et al., 2022a). To isolate the effect of using the neighborhood signal as a class target rather than as an ordering target, we keep the CAST-LN pipeline fixed and change the objective.

Both variants use the same layer-4 embeddings, $k = 10$ reciprocal neighborhoods, $N = 1{,}200$ memory, $108{,}544$ adapted parameters, optimizer, learning rate, and prequential update order as CAST-LN. The *Soft* variant trains against the mean frozen posterior of the reciprocal neighbors. The *Hard* variant thresholds that quantity at θ to form a binary target.

The Soft variant improves discrimination in all nine cells. On the two held-out Generator checkpoints, its gain is 0.031–0.041, compared with 0.062–0.067 for CAST-LN. The Hard variant is worse than the frozen detector in eight of the nine stream–checkpoint cells.

More importantly, both class-target variants move the operating point strongly toward accepting bonafide speech. Across the two variants and three main streams, 17 of 18 cells reach FAR above 92% at θ, with FRR near zero. Independent label-free recalibration does not resolve this behavior: neither class-target variant delivers lower cost than the recalibrated frozen detector in any of the 18 cells.

Table 7: Change in local minDCF relative to the frozen detector for CAST-LN and two class-target variants using the same neighborhood signal. Negative values indicate better discrimination. Checkpoint 1234 is the CAST development checkpoint.

	STATIONARY			GENERATOR			PRIOR		
Arm	1234	1235	1236	1234	1235	1236	1234	1235	1236
CAST-LN	$-.068$	$-.069$	$-.069$	$-.067$	$-.062$	$-.067$	$-.016$	$-.017$	$-.030$
Soft	$-.026$	$-.038$	$-.028$	$-.040$	$-.041$	$-.031$	$-.004$	$-.046$	$-.017$
Hard	$+.007$	$+.031$	$+.039$	$+.020$	$+.047$	$+.043$	$-.005$	$+.020$	$+.037$

The comparison separates the information in the neighborhoods from the way that information is used. A class-level loss can be reduced by moving scores collectively toward one class. In contrast, the CAST ranking loss depends only on pairwise score differences and is invariant to a uniform additive shift. Yet the class-target updates align at least as strongly with the labeled update as CAST-LN does on the shared transformer LayerNorm parameters. Across the three checkpoints, the Soft-arm cosine similarities are $+0.48$, $+0.41$, and $+0.48$, compared with $+0.27$, $+0.27$, and $+0.38$ for CAST-LN. Alignment with the labeled update is therefore not unique to the ranking formulation. The neighborhood signal already provides a supervision-like update direction; the ranking objective primarily constrains how that signal can alter relative scores.

E.3 RANK-PRESERVING OPERATING-POINT PROBE

The source-regime results show that improved local ordering does not determine a useful operating point. When CAST-LN runs on SOURCE, FAR at the deployed threshold rises from about 17% for the frozen detector to 84–89%, even though local minDCF improves. For CAST-FT, FAR reaches 96.8%, 92.0%, and 78.1% across the three checkpoints. On the checkpoint with 96.8% FAR, local minDCF still improves, directly separating discrimination from operating-point placement.

A separate labeled SOURCE diagnostic that updates the transformer trunk and AASIST back-end produces FAR values of 81.4%, 20.5%, and 75.5%. Large operating-point movement is therefore not unique to the label-free objective, although the labeled update does not produce it on every checkpoint.

To separate score location from ordering, we apply a causal monotone transformation to CAST's recorded scores. The transformation is estimated from the trailing $3{,}000$ past scores and held fixed within each 600-utterance evaluation window. It moves the adapted score distribution toward the frozen detector's score location while preserving the ordering within each window. Local minDCF is therefore unchanged within every window.

The transformation nearly restores the frozen source operating point. On one SOURCE checkpoint, for example, FAR moves from 84.8% before transformation to 17.6%, compared with 16.8% for the frozen detector on the same windows. Across checkpoints, the transformation lowers actDCF by 0.60–0.79 on SOURCE. Applied to the three main target streams, however, it raises actDCF by 0.02–0.39. Restoring a source-like score location is therefore beneficial in the source regime but generally harmful after target shift.

The transformation is a diagnostic rather than an adaptation method because it presupposes the score location to restore. The observed effect is not explained by numerical details of the mapping: at most 0.3% of scores require extrapolation beyond the fitted range, and applying the same procedure to frozen scores changes actDCF by only 0.0004–0.0027 and FAR by at most 0.3 percentage points.

E.4 ATTEMPTED LABEL-FREE OPERATING-POINT CONTROLS

We test two ways to combine CAST's discrimination improvement with label-free control of the operating point.

Score anchoring. We add a penalty for departure of the adapting detector's score location from that of the frozen detector. The weight $\lambda = 10$ was selected on one SOURCE development checkpoint

and then fixed. Anchoring preserves the source operating point on all three checkpoints and retains approximately 40–48% of CAST's discrimination gain on GENERATOR and STATIONARY.

This trade-off does not solve target operating-point identification. Across the nine stream–checkpoint cells of the three main target streams, the anchored detector delivers lower cost than the independently recalibrated frozen detector in only three cells, all from the same checkpoint. Holding the adapted scores near their source location can therefore preserve source behavior without reliably identifying an appropriate operating point after target shift.

Label-free recalibration of CAST. We also apply MLLS recalibration to the adapted detector. On SOURCE, MLLS estimates a spoofed fraction between 0.2 and 0.5, whereas the true fraction is approximately 0.9, and FAR remains between 46% and 77%. The adapted score distribution therefore violates the conditions under which the class-prior estimate would restore the source operating point.

Together with the rank-preserving transformation, these controls separate two requirements of a label-free continual detector. Preserving a known source score location can protect source behavior but does not identify the appropriate location after target shift. Conversely, estimating the target prior from adapted scores can be unreliable when adaptation has itself changed their distribution. Discrimination repair and operating-point identification therefore remain separate problems in this benchmark.

The SOURCE quantities in this appendix measure behavior while a method is run directly on the source-regime stream. They are distinct from post-target source retention, where SOURCE is rescored after adaptation on a target stream; the latter is the measurement on which CAST-LN degrades with WavLM without labeled replay.

F DETAILED RESULTS AND ROBUSTNESS

F.1 PER-CHECKPOINT XLS-R RESULTS

Table 8 gives the per-checkpoint changes in local minDCF underlying the means in Table 2. Checkpoint 1234 was used for CAST development on GENERATOR, so the reported CAST mean on that stream uses checkpoints 1235 and 1236; checkpoint 1234 is shown in the table for completeness. Every comparison is made within checkpoint against the corresponding frozen detector. Uncertainty is estimated with the paired bootstrap described in Appendix A.5.

Table 8: Per-checkpoint change in local minDCF relative to the frozen detector on the three main XLS-R target streams. Negative values indicate better discrimination. Checkpoint 1234 was used for CAST development on GENERATOR.

	STATIONARY			GENERATOR			PRIOR		
Method	1234	1235	1236	1234	1235	1236	1234	1235	1236
Tent	+.0034	+.0188	−.0060	−.0034	−.0127	−.0101	−.0024	+.0101	−.0149
EATA-rd	−.0005	+.0005	−.0037	−.0011	−.0062	−.0020	−.0014	−.0045	−.0038
SAR	+.0001	+.0009	−.0007	+.0004	−.0007	−.0004	−.0009	+.0007	+.0000
RoTTA	+.0113	+.0849	+.0059	+.0125	+.0706	+.0020	+.0196	+.2818	+.0122
RDumb	−.0005	+.0002	−.0032	−.0014	+.0073	+.0010	−.0030	+.0026	−.0049
CAST-LN	−.0677	−.0691	−.0694	−.0670	−.0617	−.0674	−.0164	−.0166	−.0301
CAST-FT	−.1126	−.1202	−.1110	−.1065	−.1010	−.1056	−.0900	−.1112	−.0922
Labeled-LN	−.1090	−.1028	−.0895	−.1101	−.1125	−.1000	−.0888	−.0821	−.0699
Labeled-FT	−.3907	−.3925	−.4110	−.3771	−.3831	−.3995	−.3994	−.4030	−.4175

The baseline effects are small and vary across checkpoints. Tent, for example, changes sign across checkpoints on both STATIONARY and PRIOR, while RoTTA's degradation on PRIOR ranges from +0.012 to +0.282. In contrast, both CAST configurations improve minDCF on every checkpoint of each main target stream. The paired bootstrap interval lies below zero for every CAST and labeled-reference cell on these three streams.

Table 9 reports the corresponding 95% paired-bootstrap intervals for the claim-bearing arms. All 36 intervals lie below zero.

Table 9: 95% paired-bootstrap intervals for the change in local minDCF relative to the frozen detector. Intervals are computed independently within each checkpoint. Checkpoint 1234 is the CAST development checkpoint on GENERATOR.

Arm	Checkpoint	STATIONARY	GENERATOR	PRIOR
CAST-LN	1234	$[-.0696, -.0627]$	$[-.0701, -.0618]$	$[-.0213, -.0105]$
	1235	$[-.0722, -.0648]$	$[-.0649, -.0565]$	$[-.0233, -.0094]$
	1236	$[-.0726, -.0648]$	$[-.0715, -.0625]$	$[-.0360, -.0237]$
CAST-FT	1234	$[-.1154, -.1054]$	$[-.1101, -.0998]$	$[-.0968, -.0814]$
	1235	$[-.1239, -.1132]$	$[-.1048, -.0945]$	$[-.1203, -.1027]$
	1236	$[-.1145, -.1048]$	$[-.1098, -.0989]$	$[-.0994, -.0845]$
Labeled-LN	1234	$[-.1104, -.1026]$	$[-.1139, -.1046]$	$[-.0938, -.0833]$
	1235	$[-.1043, -.0964]$	$[-.1164, -.1066]$	$[-.0876, -.0768]$
	1236	$[-.0920, -.0842]$	$[-.1030, -.0946]$	$[-.0750, -.0641]$
Labeled-FT	1234	$[-.3866, -.3725]$	$[-.3785, -.3638]$	$[-.4071, -.3870]$
	1235	$[-.3891, -.3747]$	$[-.3843, -.3691]$	$[-.4116, -.3909]$
	1236	$[-.4075, -.3923]$	$[-.4010, -.3854]$	$[-.4260, -.4056]$

Among the prediction-driven baselines, Tent's interval lies below zero on all three GENERATOR checkpoints: $[-.00630, -.00002]$, $[-.01589, -.00834]$, and $[-.01372, -.00672]$. EATA-rd's interval lies below zero on two of three PRIOR checkpoints: $[-.00385, +.00016]$, $[-.00690, -.00142]$, and $[-.00708, -.00212]$. No other prediction-driven baseline has intervals below zero on more than one checkpoint of any main target stream.

The matched Tent comparison shows how much larger the available supervised discrimination gain is. On GENERATOR, Tent changes minDCF by -0.003, -0.013, and -0.010 across the three checkpoints, whereas Labeled-LN changes it by -0.110, -0.112, and -0.100. The discrimination improvements obtained by the prediction-driven baselines are therefore small relative to those available when correct target labels are supplied.

F.2 RECOVERY OF CALIBRATION LOSS

We quantify how much of the frozen detector's calibration loss can be removed without changing its score ordering. Let

$$\rho = 1 - \frac{C^{\text{recal}}}{C^{\text{frozen}}}, \qquad C = \text{actDCF} - \text{minDCF},$$

where C is computed locally as in Appendix A.5 and averaged over the corresponding blocks or windows before taking the ratio. A value of $\rho = 1$ means that recalibration removes the calibration loss completely.

Table 10: Fraction ρ of the frozen detector's calibration loss removed by label-free MLLS recalibration. Recalibration changes the operating threshold but not the detector or its score ordering.

	XLS-R			WavLM		
Stream	1234	1235	1236	1234	1235	1236
STATIONARY	0.75	0.29	0.44	0.56	0.49	0.69
GENERATOR	0.68	0.28	0.42	0.42	0.38	0.57
PRIOR	0.77	0.22	0.43	0.63	0.54	0.76

MLLS recovers 22–77% of the calibration loss on XLS-R and 38–76% on WavLM without changing discrimination. Recovery varies substantially across source checkpoints. MLLS assumes pure label shift, $p_t(x \mid y) = p_s(x \mid y)$ (Lipton et al., 2018), whereas the cross-corpus target shift also changes the class-conditional distributions. The pure-label-shift assumption therefore does not hold in this benchmark. Applying bias-corrected temperature scaling before MLLS (Alexandari et al., 2020) does not improve recovery.

F.3 NEIGHBORHOOD AND DIRECTION CONTROLS

Table 11 isolates two components of the CAST signal on GENERATOR. Setting $k = 0$ removes neighborhood information while retaining the adaptation procedure. Reversing the pseudo-score ordering retains the neighborhood relation but reverses the direction supplied to the ranking objective.

Table 11: Controls for CAST-LN on GENERATOR: change in local minDCF relative to the frozen detector. Negative values indicate better discrimination. Checkpoint 1234 is the development checkpoint.

Variant	1234	1235	1236
CAST-LN	$-.067$	$-.062$	$-.067$
Own score only ($k = 0$)	$+.013$	$+.017$	$+.007$
Reversed target order	$+.493$	$+.513$	$+.488$

Removing neighborhood information worsens discrimination on all three checkpoints. Reversing the target ordering produces a much larger degradation and raises FAR at θ to 99.6–99.9%. The improvement obtained by CAST therefore depends on both the neighborhood relation and the direction of the ordering signal; it is not explained merely by taking an adaptation step.

F.4 METRIC ROBUSTNESS

We compute threshold-optimized metrics locally because an adapting detector can move its score scale over time. Pooling scores across such regions can therefore combine within-region discrimination with between-region score movement. This distinction matters empirically: across the 133 evaluated arm–checkpoint–stream cells, local and pooled ΔEER have opposite signs in 35. Pooled EER can therefore change even when the direction of the local discrimination change is opposite.

Local EER and local minDCF agree more closely. Their mean changes have opposite signs in 19 of the same 133 cells, and 14 of those 19 have $|\Delta\mathrm{minDCF}| < 0.005$. None of the headline conclusions changes: both metrics improve for CAST-LN, CAST-FT, and the labeled references on every checkpoint of STATIONARY, GENERATOR, and PRIOR. We use minDCF as the primary discrimination measure because the decomposition in equation 1 is defined directly in terms of it.

F.5 WAVLM TRANSFER AND SOURCE RETENTION

The CAST-LN discrimination gain transfers to WavLM. Mean local minDCF decreases by 0.022 on STATIONARY, 0.057 on GENERATOR, and 0.012 on PRIOR. On GENERATOR, the paired interval lies below zero on all three WavLM checkpoints. These experiments use the CAST configuration selected with XLS-R; no WavLM result was used for development.

The prediction-driven baselines behave differently. At the matched $110{,}594$-parameter surface, Tent, EATA-rd and RDumb change per-block minDCF by at most 0.0012 on any of the three target streams and move FAR at θ by less than one point, against changes an order of magnitude larger on XLS-R. This is an update that barely happens rather than one that happens and fails: EATA-rd's relative parameter displacement is 1.6 to 2.3×10^{-5} on WavLM against 7.1×10^{-5} to 2.0×10^{-4} on XLS-R at the identical surface, Tent's is damped by a similar factor, and the ratio between the two arms is preserved across backbones, so the damping is a property of the backbone rather than of any one objective. The labeled reference on the same surface shows that the surface is movable: it improves minDCF by 0.051 to 0.078 and moves FAR from 81.2% to between 54.8% and 72.0%, which is 70 to 75% of its XLS-R gain. The label-free objectives are not finding gradient there. These are point estimates over three checkpoints. No learning rate was swept on WavLM for any arm, so whether a larger rate would change this is not answered here.

The limitation appears when source performance is measured after target adaptation. Table 12 reports the change in source minDCF after CAST-LN traverses each WavLM target stream. After GENERATOR and PRIOR, source minDCF is higher than for the frozen detector on all three checkpoints. After STATIONARY, one checkpoint shows a large degradation while the other two show small improvements.

Table 12: WavLM source retention after CAST-LN adapts on each target stream. Values are changes in source minDCF relative to the frozen detector; negative values indicate better source discrimination. Labeled source replay is a diagnostic modifier and is not part of the source-free CAST method.

	No replay			Labeled source replay		
Adapted on	1234	1235	1236	1234	1235	1236
STATIONARY	$+0.188$	-0.023	-0.022	-0.035	-0.007	-0.015
GENERATOR	$+0.107$	$+0.241$	$+0.071$	-0.043	-0.035	-0.065
PRIOR	$+0.144$	$+0.146$	$+0.103$	-0.027	-0.029	-0.031

Adding labeled source replay prevents the source degradation in all nine cells; source minDCF is below the frozen value in each case. Replay also reduces the target discrimination gain: on GENERATOR, the mean improvement falls from 0.057 without replay to 0.046 with replay. Because replay requires labeled source data, we use it only as a diagnostic showing that the WavLM retention failure can be counteracted when source supervision is available. It is not part of CAST.

CAST-FT and the full-trunk labeled reference were not run on WavLM, so the claims for those two arms are scoped to XLS-R throughout. SOURCE exists only for XLS-R, so the retention measurements above use the source partition re-scored after adaptation rather than a WavLM source-regime stream.