

Audio-JEPA Representations for Speech Deepfake Detection

Nguyen Le Nguyen^{1,2} and Tran Van Hoai^{1,2}

¹ Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City, Vietnam

² Vietnam National University Ho Chi Minh City (VNU-HCM), Ho Chi Minh City, Vietnam
{nlnghuyen.sdh231,hoai}@hcmut.edu.vn

Abstract. Speech deepfake detectors are built on self-supervised front-ends, but published systems differ in architecture, size and pre-training corpus at once, so the contribution of the pre-training objective itself is hard to read off. We apply Audio-JEPA — a ViT-Base encoder pre-trained on AudioSet to predict latent representations of masked spectrogram patches — combining its layer outputs by learned weights, attentive statistics pooling and a small classification head. We train once on the ASVspoof2019 LA training partition with a fixed recipe; evaluation runs on the disjoint evaluation partition and further corpora through the Speech DF Arena toolkit, against verified published baselines. Under an identical downstream recipe we compare it against AudioMAE, a released checkpoint that shares its ViT-Base backbone and its AudioSet pre-training corpus but is trained to reconstruct masked patches rather than to predict them: latent prediction is ahead in three of the four comparisons, by 2.57 to 7.55 EER points over three seeds per arm with no overlap between seed ranges. With the encoder frozen, the system reaches 8.47 % EER on DFADD, where our re-run of the AASIST baseline gives 41.87 % and the best system trained on the same corpus, the 340 M-parameter XLSR+SLS, gives 7.54 %. Performance varies across evaluation corpora, with lower error rates on those sharing the training set’s bona-fide source; we discuss this pattern and its implications for future pre-training data choices

Keywords: speech deepfake detection · speech anti-spoofing · self-supervised learning · representation learning · joint-embedding predictive architecture · cross-corpus generalization

1 Introduction

Detecting synthetic or converted speech — the *logical access* spoofing task of the ASVspoof challenges [28] — has become a problem of generalisation rather than of in-domain accuracy. On the ASVspoof2019 LA evaluation partition, ten of the twelve open-source systems collected by the Speech DF Arena benchmark [8] are below 5 % equal error rate and the best is at 0.12 %, but on corpora recorded under other conditions or attacked by newer generators the same systems range from a few percent to near chance. Every system near the top of that leaderboard is built on a self-supervised speech front-end: wav2vec 2.0 and XLS-R [3,2], trained with a contrastive objective on

hundreds of thousands of hours of speech, or WavLM and HuBERT [5,12], trained to predict masked discrete units.

One option is absent from that leaderboard: a pre-training target that is continuous and produced by the model itself. Joint-embedding predictive architectures [16,1,4] learn by predicting the *representation* of a masked region from its context, with the target supplied by an exponential moving average of the encoder being trained rather than by a codebook, a set of negatives or the input signal; Audio-JEPA [30] recently applied the recipe to spectrograms and released a ViT-Base encoder pre-trained on AudioSet. Among the twelve open-source systems in the Speech DF Arena paper, no front-end is pre-trained this way, and none is pre-trained to reconstruct its input either; the closest prior work we know of evaluated AudioMAE features for deepfake detection and found them the weakest of the deep features it compared, attributing the result to the AudioSet pre-training corpus [35].

Which property of a front-end matters is hard to read off the leaderboard, because the systems on it differ in architecture, parameter count, pre-training corpus and input representation at the same time. This paper applies Audio-JEPA to speech deepfake detection with a light back-end, trained once on ASVspoof2019 LA with a fixed recipe and evaluated on five public corpora through the Speech DF Arena toolkit, after checking that toolkit against the published baseline numbers. It then places that result beside AudioMAE [13], the closest available checkpoint — same backbone, same pre-training corpus, a reconstruction objective instead of a predictive one — run through an identical downstream recipe.

Latent prediction leads masked reconstruction by 2.57 to 7.55 EER points in three of the four comparisons we ran, across two corpora and both adaptation regimes, with no overlap between seed ranges.

2 Background

Self-supervised front-ends for anti-spoofing. Tak et al. [27] showed that a fine-tuned wav2vec 2.0 XLS-R encoder feeding the AASIST graph-attention back-end [14], trained with RawBoost augmentation, generalised far better than end-to-end models trained from scratch; Wang and Yamagishi [31] compared several speech SSL front-ends and investigated back-end and fine-tuning choices, likewise finding benefits from SSL front-ends. The systems that lead the Speech DF Arena leaderboard among those trained on ASVspoof2019 — XLSR+SLS [38], TCM [29], XLSR-Mamba [33], Nes2Net [18] — all keep an XLS-R front-end of about 300 M parameters and vary the back-end. Other evaluated SSL systems use WavLM, HuBERT and wav2vec 2.0 encoders with an ECAPA-TDNN [6] back-end. All of these front-ends are speech models, pre-trained on speech corpora ranging from the 960 hours of LibriSpeech for the base HuBERT and WavLM checkpoints to the 436 000 hours behind XLS-R.

Where the pre-training target comes from. Self-supervised objectives can be told apart by what they ask the encoder to predict and by what supplies the answer. *Contrastive* objectives (wav2vec 2.0) identify the true quantised target of a masked frame among distractors drawn from the same utterance. *Masked-unit prediction* (HuBERT, WavLM)

predicts cluster identities assigned offline by k -means. *Masked reconstruction* (AudioMAE [13]) decodes masked spectrogram patches back to their values and minimises the error in input space. *Latent prediction* (I-JEPA, Audio-JEPA) uses continuous targets supplied by an exponential moving average of the encoder: with a context encoder f_θ , a predictor g_ϕ and a target encoder $f_{\bar{\theta}}$ updated as an exponential moving average of f_θ , it minimises

$$\mathcal{L} = \frac{1}{|M|} \|g_\phi(f_\theta(x_{\text{ctx}}), m) - \text{sg}[f_{\bar{\theta}}(x)]_M\|_2^2, \quad (1)$$

where m encodes the positions of the masked target patches and sg stops gradients. The target encoder processes the full input x ; its outputs at the masked positions M provide the targets. Latent prediction may allow representations to omit unpredictable input details; whether this occurs and helps or hurts synthesis-artefact detection remains an empirical question.

Speech DF Arena. Dowerah et al. [8] released an evaluation toolkit and a leaderboard covering fourteen deepfake corpora. Their paper reports twelve open-source systems, all trained on ASVspoof2019 LA (Whisper-MesoNet [15] excepted), with released per-utterance score files; this is what makes an independent re-run checkable. We evaluate on five of the fourteen corpora (Sect. 4.2), chosen to span genuine-speech sources and spoofing types.

3 Method

3.1 Front-end: Audio-JEPA

Audio-JEPA [30] applies the joint-embedding predictive architecture of I-JEPA [1] to log-mel spectrograms. During pre-training a context encoder sees a subset of spectrogram patches, and a small predictor is trained to output the representations that a target encoder — an exponential moving average of the context encoder — assigns to masked patches. The loss is computed in representation space; no patch is ever reconstructed. The published checkpoint³ follows the ViT-Base configuration [7] (12 blocks, width 768, 12 heads, 85.4 M parameters) and is pre-trained on AudioSet-2M [11]: 5 338 hours of general audio spanning speech, music and environmental sound [30]. We use the context encoder and do not use the target encoder or the predictor.⁴

The toolkit hands the model a 16 kHz waveform tiled or truncated to 64 600 samples (Sect. 4.1). We resample to 32 kHz, following the Audio-JEPA convention — this adds no information, since the source is 16 kHz — take the first 2.56 s (tiling if shorter), and compute a 128-band log-mel spectrogram with a 25 ms window, 10 ms hop and a 20–8 000 Hz range, after subtracting the waveform mean; the spectrogram is zero-padded or truncated to 256 frames. The 256×128 spectrogram is cut into patches of 8 frames by 32 mel bins (eight 10 ms frame steps $\times$ a quarter of the mel axis), giving

³ <https://huggingface.co/ltuncay/Audio-JEPA>

⁴ Both I-JEPA and Audio-JEPA evaluate with the target encoder, which is an exponential moving average of the context encoder. We use Audio-JEPA’s context encoder; the effect of choosing it rather than its target encoder was not measured.

a grid of $32 \times 4 = 128$ tokens, each projected linearly to 768 dimensions and given a two-dimensional sinusoidal position embedding; there is no class token. The pre-trained patch embedding is 16×16 ; we interpolate its weights bicubically to 8×32 and regenerate the position embeddings for the new grid. The token count is unchanged by this: Audio-JEPA’s 16×16 patches over a 128×256 spectrogram give 128 patches, and our $(8, 32)$ patches over 256×128 give 32×4 , also 128. These counts refer to the full patch grids before masking.

These input settings differ from the pre-training recipe, which uses 10 s clips at 32 kHz with 128 mel bands over 256 frames [30] — about a 39 ms hop, four times coarser in time than ours, with the released configuration setting the upper frequency to 16 kHz.⁵ The released configuration operates at a ten-second scale. In an exploratory comparison, it reduced EER on the held-out attack subset by about a factor of five but increased evaluation-partition EER; these observations informed our decision to retain the finer-resolution configuration.

3.2 Back-end

The back-end is deliberately conventional. We aggregate the encoder’s layer outputs using learned weights and summarise the resulting token sequence with attentive statistics pooling [21]. Both arms of Sect. 3.4 use this back-end unchanged.

The encoder yields thirteen token sequences of shape 128×768 : the output of each of the twelve blocks and the final layer-normalised output. Three steps turn them into a score. A learned softmax over thirteen scalars weights the sequences and sums them (13 parameters, initialised to zero and hence uniform after the softmax). Attentive statistics pooling [21] — a two-layer attention network ($768 \rightarrow 128 \rightarrow 1$, tanh) with a softmax over the 128 tokens — produces attention-weighted mean and standard deviation vectors, concatenated to 1 536 dimensions (98 561 parameters). A classifier of LayerNorm, Linear $1\,536 \rightarrow 256$, GELU, Dropout 0.2 and Linear $256 \rightarrow 2$ gives two logits (397 058 parameters). The back-end totals 495 632 parameters, about 0.5 M, or 0.58 % of the encoder. The score written for each utterance is the logit difference $d = z_{\text{bonafide}} - z_{\text{spoof}}$; we use the difference rather than a single logit because the softmax probability is monotone in the difference, not in either logit alone, and the toolkit’s EER depends only on score order.

Figure 1 distinguishes the published pre-training procedure from the downstream pipeline used here.

3.3 Training

We train in two regimes. *Frozen*: the encoder runs in evaluation mode without gradients and only the back-end is trained. *Fine-tuned*: both encoder and back-end are fine-tuned, with a peak learning rate of 3×10^{-5} for the encoder. The back-end uses a peak learning rate of 10^{-3} in both regimes. Both use AdamW [19] with weight decay 0.05, a one-cycle schedule with 10 % warm-up, batch size 32, mixed precision and exactly six epochs. The loss is cross-entropy with the bona fide class up-weighted by the spoof-to-bona-fide

⁵ <https://github.com/LudovicTuncay/Audio-JEPA>

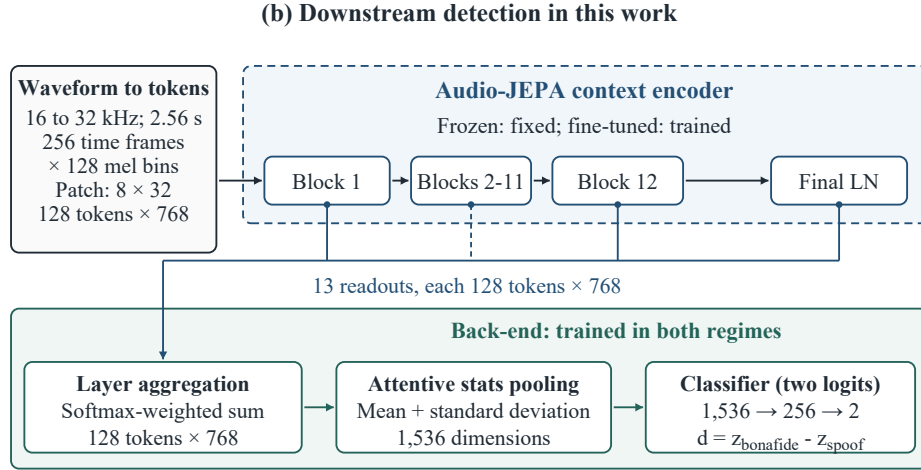
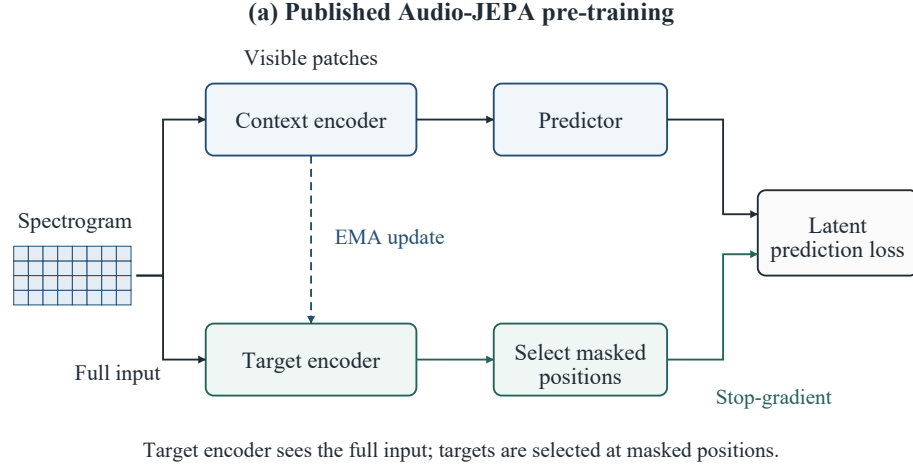


Fig. 1. (a) Audio-JEPA pre-training, schematically redrawn from [30]: the context branch predicts representations at masked positions; the target encoder sees the full input and is updated by exponential moving average (EMA). Gradients do not pass through the target branch. (b) Downstream detection using the released context encoder; we do not repeat pre-training. Its twelve block outputs and final layer-normalised output are aggregated. Only the back-end is trained in the frozen regime; both encoder and back-end are trained in the fine-tuned regime.

ratio of the training set (about 9.6). The only augmentation, applied during training, is a SpecAugment-style [22] mask drawn afresh for each spectrogram: one band of up to 16 of the 256 frames and one of up to 16 of the 128 mel bins, at random positions — at most 6.25 % of the time axis and 12.5 % of the mel axis. This differs from the patch masking used during encoder pre-training, which masks 40–60 % of patches to define

the latent-prediction task. Training data and hold-out are described in Sect. 4.2; the final six-epoch checkpoint is used for each main configuration (Sect. 4.1).

Every configuration in Tables 2 and 4 is trained with three seeds, 1234, 2 and 3, except where a single seed is marked. We report both the frozen and the fine-tuned regime, not to choose between them but because they give that comparison a second axis: a difference that survives with the encoder frozen cannot be an artefact of how fine-tuning interacts with one checkpoint or the other.

3.4 Comparison against a reconstruction-trained checkpoint

The closest available point of comparison for Audio-JEPA is AudioMAE [13]: the same ViT-Base backbone, the same AudioSet-2M pre-training corpus, and a reconstruction objective in place of a predictive one. We take the encoder of the public checkpoint (the `timmpart vit_base_patch16_1024_128.audiomae_as2m`), interpolate its patch embedding to 8×32 in the same way, and give it the same input, back-end, training data, recipe and seeds.

The checkpoints share the ViT-Base architecture and AudioSet-2M as their pre-training data source, and we use a common downstream recipe. Their pre-training recipes nevertheless differ in masking ratio, sample rate, optimiser and compute budget, among other factors. The two encoders are also displaced from their pre-training input configuration along different axes: our 2.56 s input shortens the clip about fourfold for both, preserves Audio-JEPA’s full patch count while quartering AudioMAE’s, and changes the nominal temporal patch span (patch length in frames multiplied by frame hop) from 625 to 80 ms for Audio-JEPA and from 160 to 80 ms for AudioMAE. No single configuration is native to both, and giving each its own would change the input alongside the objective. We therefore read the comparison as one between two released checkpoints under a common downstream recipe, and treat the attribution of the difference to the objective itself as a hypothesis, examined in Sect. 6 and bounded in Sect. 8.

4 Experimental Setup

4.1 Evaluation toolkit and protocol

All evaluations use the Speech DF Arena toolkit [8]. In the fixed-length configuration we run, the toolkit loads every file at 16 kHz, tiles or truncates it to 64 600 samples (about 4 s), scores it with the model under test and computes the equal error rate (EER) from the resulting bona fide and spoof score lists. Our systems are wrapped as toolkit models, and the score written for each utterance is the logit difference $d = z_{\text{bonafide}} - z_{\text{spoof}}$, so that a higher score means bona fide, which is the convention the toolkit assumes. We use the toolkit’s own EER routine throughout and report EER in percent; lower is better. Where a configuration was trained with three seeds we report the mean and the sample standard deviation over seeds and, where it matters, the per-seed range.

For each main Audio-JEPA and AudioMAE configuration, we evaluate the final six-epoch checkpoint without early stopping or performance-based checkpoint selection. During these runs, the A05 hold-out is monitored but does not determine which checkpoint is evaluated. Exploratory variants were scored on evaluation data and informed configuration choices. Both encoders use the same final downstream recipe.

Table 1. Evaluation corpora, and the AASIST baseline re-run through our pipeline against the figure published in Table 2 of [8] (EER in %).

Corpus	Utterances	Genuine	Spoofed speech	AASIST	
				Ours	Publ.
ASVspoof2019 LA eval [28]	71 237	VCTK	13 TTS/VC attacks (11 unseen)	0.83	0.82
DFADD [9]	3 755	VCTK	5 diffusion / flow TTS systems	41.87	41.86
ADD 2022 Track 1 [36]	109 199	Mandarin	noisy, low-quality fakes	47.92	47.91
LibriSeVoc [24]	18 487	LibriTTS	6 neural vocoders	37.95	37.65
In-the-Wild [20]	31 779	web sources	fakes collected from the internet	43.02	43.00

4.2 Datasets

Training uses the ASVspoof2019 LA training partition [28,32]: 25 380 utterances (2 580 bona fide, 22 800 spoofed) from 20 speakers and six attack algorithms, A01–A06: four text-to-speech systems and two voice conversion systems. A random utterance split does not enforce separation by speaker or attack algorithm, so we instead hold out one attack algorithm. We hold out A05 — voice conversion with a WORLD vocoder — together with 600 bona fide utterances, leaving 20 980 utterances for training. Its vocoder is shared with A02 and A03 and its family with A06. In the main training runs, A05 is monitored without being used for checkpoint selection; excluding it changes the training partition. The bona fide speech of ASVspoof2019 is drawn from the VCTK corpus [34].

Evaluation uses the five corpora in Table 1, all scored through the toolkit’s protocol files. They differ in the origin of their genuine speech as well as in the kind of spoofing, and both distinctions matter for reading the results.

ASVspoof2019 LA evaluation partition (7 355 bona fide, 63 882 spoofed) contains thirteen attacks, A07–A19; two of them (A16, A19) reuse the algorithms of training attacks A04 and A06, the other eleven are unseen. *DFADD* (755 bona fide, 3 000 spoofed) pairs VCTK speech with five recent diffusion- and flow-matching-based TTS systems, 600 utterances each; we use the copy distributed through HuggingFace⁶. *ADD 2022 Track 1* (31 334 bona fide, 77 865 spoofed) is Mandarin, low quality, and contains real-world noise and background music. *LibriSeVoc* re-synthesises LibriTTS [37] utterances with six neural vocoders, so its spoofed speech is clean vocoder output on read speech. *In-the-Wild* (19 963 bona fide, 11 816 spoofed) collects genuine and fake recordings of public figures from the internet.

4.3 Baseline re-run and pipeline verification

Before reporting our own numbers we checked the pipeline in three ways.

⁶ <https://huggingface.co/datasets/SpeechAntiSpoofingBenchmarks/DFADD>

First, the leaderboard releases per-utterance score files for its systems. Recomputing EER from those files with our own label files reproduces Table 2 of [8] to within 0.01 point for seven of the eight released score sets we checked on ADD 2022 Track 1; this checks the metric and our label mapping without touching any audio.

Second, we ran the official AASIST checkpoint [14] through our installation of the toolkit on all five corpora (last two columns of Table 1), and AASIST is re-run alongside every one of our own evaluations. On ASVspoof2019 LA, ADD 2022 Track 1 and In-the-Wild the re-run matches the published figure to within 0.02 point; on LibriSeVoc it is within 0.30 of the figure in the paper and within 0.07 of the current online leaderboard value (38.02). On DFADD, AASIST have a small difference by 0.01.

Third, for each checkpoint we score the same waveform once through the toolkit wrapper and once directly in the training code and require the two scores to agree to 10^{-3} , which checks that the wrapper reproduces the trained model.

4.4 Comparison conditions

To limit differences in downstream training data, we draw published baseline results from the ASVspoof2019 LA-trained systems in Table 2 of the Speech DF Arena paper [8], while retaining Whisper-MesoNet, which uses different training data, for context only. Our systems use the same source corpus but exclude A05 and 600 bona fide utterances for monitoring (Sect. 4.2), so the training subsets are not identical. We also report our own AASIST re-run. These are system-level comparisons: architectures, pre-training data and training recipes differ. The online leaderboard [23] is cited for context only, with its access date, since its contents change.

5 Results

5.1 Five benchmarks

Table 2 gives EER on the five corpora for the frozen and the fine-tuned Audio-JEPA system alongside the AASIST re-run.

The clearest result is on DFADD. With the encoder frozen and only the back-end trained, the system reaches 8.47 % (seeds 8.74, 8.59, 8.07) against 41.87 % for AASIST. Fine-tuning yields a higher mean EER of 10.75 % and approximately ten times the sample standard deviation across seeds (7.57, 10.20, 14.47).

On the ASVspoof2019 LA evaluation partition the picture reverses. Fine-tuning brings the system from 17.70 to 7.39 %, against 0.83 % for AASIST. This is an in-corpus evaluation, on a partition disjoint from training.

On ADD 2022 Track 1 the fine-tuned system is 3.8 points better than AASIST (44.10 against 47.92), in a corpus where the ASVspoof2019-trained systems in the arena paper span 31.04 to 50.35 %. On LibriSeVoc and In-the-Wild, the reported Audio-JEPA results have higher EER than AASIST in both regimes. The frozen In-the-Wild results vary substantially across seeds, from 37.74 to 53.39 %, spanning values both below and above the AASIST baseline.

The two corpora on which the system reaches its lowest EERs, DFADD and ASVspoof2019, are the two whose genuine speech comes from VCTK, the same corpus

Table 2. EER (%) on five corpora. Audio-JEPA figures are mean $\pm$ sample standard deviation over three seeds unless marked; the frozen system trains only the 0.50 M-parameter back-end, the fine-tuned system updates all 85.4 M encoder parameters as well. AASIST is the official checkpoint re-run through the same pipeline.

Corpus	Genuine speech	JEPA frozen (0.5 M trainable)	JEPA fine-tuned (85.9 M trainable)	AASIST (0.3 M)
DFADD	VCTK	8.47 $\pm$ 0.35	10.75 $\pm$ 3.48	41.87
ASVspoof2019 LA eval	VCTK	17.70 $\pm$ 1.44	7.39 $\pm$ 0.69	0.83
ADD 2022 Track 1	Mandarin	—	44.10 $\pm$ 2.04	47.92
LibriSeVoc	LibriTTS	47.38 $\pm$ 4.19	41.96 $\pm$ 2.93	37.95
In-the-Wild	web sources	47.55 $\pm$ 8.55	52.00 [†]	43.02

[†]one seed. — not run.

as the training data; the three corpora whose genuine speech comes from elsewhere — LibriTTS, the internet, Mandarin speakers — are where it is weakest. Sect. 6 returns to this.

We provide a detailed comparison with published systems on DFADD, where our system achieves its lowest EER among the five evaluation sets; this result does not represent its performance across corpora. Table 3 includes the top eleven open-source systems reported in the arena paper and our two configurations.

5.2 Audio-JEPA versus AudioMAE under a common downstream recipe

Table 4 gives the four comparisons we ran.

Audio-JEPA is ahead in three of the four comparisons, by 2.57 to 7.55 points, and in each of the three the seed ranges of the two encoders do not overlap: two corpora, two adaptation regimes, the same direction. In the fourth comparison, ADD 2022 Track 1 with fine-tuning, AudioMAE has the lower mean, by 2.08 points, with overlapping seed ranges.

On ADD 2022 Track 1 the first seed alone puts Audio-JEPA ahead by 0.67 points, whereas the three-seed means put AudioMAE ahead by 2.08. We therefore report both mean differences and per-seed ranges. Non-overlapping ranges serve only as a descriptive criterion, not a test of statistical significance; three comparisons meet this criterion and one does not. ADD 2022 Track 1 is the noisy, low-quality corpus among the three datasets, whereas the three comparisons with non-overlapping ranges involve clean speech. Sect. 6 offers a hypothesis for this pattern.

6 Discussion

One account of the difference between the checkpoints. Masked autoencoding minimises reconstruction error in input space. Under squared loss, the optimal prediction is the conditional expectation given the context, so uncertain input details can be averaged out in the reconstruction. This does not establish that those details are absent from the encoder’s representations. One hypothesis is that some spectral details useful

Table 3. DFADD results from Table 2 of [8] and our systems: All systems trained on ASVspoof2019 LA except Whisper-MesoNet, and ours, sorted by EER (lower is better). Parameter counts as reported by the arena paper.

#	System	Params	EER (%)
1	XLSR+SLS [38]	340 M	7.54
2	Audio-JEPA frozen (ours)	85.9 M*	8.47
3	TCM [29]	319 M	8.88
4	XLSR-Mamba [33]	319 M	10.69
5	Audio-JEPA fine-tuned (ours)	85.9 M	10.75
6	Nes2Net [18]	318 M	11.14
7	wav2vec2-AASIST [27]	318 M	11.92
8	RawGAT-ST [25]	0.44 M	23.70
9	Whisper-MesoNet [15] [‡]	7.6 M	24.11
10	RawNet2 [26]	17.6 M	26.19
11	WavLM-ECAPA [5,6]	102 M	29.53
12	HuBERT-ECAPA [12,6]	102 M	34.56
13	AASIST [14]	0.30 M	41.87

*0.5 M of them trained. [‡]Not trained on ASVspoof2019 according to [8].

Table 4. Comparison of two released checkpoints under an identical downstream recipe. EER (%), mean $\pm$ s.d. over three seeds per arm, with per-seed ranges. Gap is AudioMAE minus Audio-JEPA; positive favours Audio-JEPA.

Corpus	Encoder	JEPA	MAE	Gap	JEPA range	MAE range
ASV2019 LA	fine-t.	7.39 $\pm$ 0.69	13.25 $\pm$ 0.57	+5.87	6.78–8.13	12.61–13.70
ASV2019 LA	frozen	17.70 $\pm$ 1.44	25.25 $\pm$ 0.68	+7.55	16.18–19.04	24.47–25.74
DFADD	frozen	8.47 $\pm$ 0.35	11.04 $\pm$ 0.75	+2.57	8.07–8.74	10.18–11.54
ADD22 Track 1	fine-t.	44.10 $\pm$ 2.04	42.02 $\pm$ 2.85	−2.08	41.75–45.29	39.00–44.65

for detecting synthesis artefacts are poorly predictable from context, and that the two pre-training objectives encourage different retention of such information. Neither the predictability of these details nor their retention by the encoders was measured here. Latent prediction operates on learned representations rather than input values; it does not require reconstruction, but this alone does not guarantee better preservation of synthesis artefacts.

The observed results are compatible with this hypothesis but do not establish it. Audio-JEPA has lower mean EER by 5.87 and 7.55 points on ASVspoof2019 LA and by 2.57 points on DFADD. On noisy ADD 2022 Track 1, AudioMAE has the lower mean by 2.08 points, with overlapping seed ranges. Noise could obscure discriminative spectral detail, but these datasets also differ in speech source, language and spoofing systems. Moreover, the checkpoints differ in more than their objective (Sect. 3.4). Controlled perturbations and representation analyses would be needed to test the proposed mechanism.

Association with the source corpus. DFADD and ASVspoof2019 draw their bona fide speech from VCTK, as does our training data, and these are the two corpora with the lowest reported Audio-JEPA EERs. The three corpora with other genuine-speech sources have higher errors. This is an association, not a causal decomposition, and a common downstream recipe does not eliminate encoder–corpus interactions.

Shared source data alone also does not guarantee low error: AASIST is trained on VCTK bona fide speech but reaches 41.87 % EER on DFADD. This does not rule out a contribution from source similarity to Audio-JEPA’s result. LibriSeVoc contains clean vocoder output but yields high Audio-JEPA error on a different speech source, showing that clean audio alone does not ensure good performance.

Directions for pre-training and downstream data. The leading speech front-ends in our comparison are pre-trained on tens to hundreds of thousands of hours of speech, whereas Audio-JEPA uses 5 338 hours of general audio. Pre-training domain and scale are therefore factors worth testing, rather than established explanations or remedies for the observed gaps. Audio-JEPA’s published X-ARES evaluation also reports stronger performance on environmental sound and music than on speech [30], motivating investigation of speech-focused pre-training.

Downstream data diversity and augmentation are separate factors not varied systematically here. Teffic-Audio [17], for example, describes a recipe combining multi-source data, balanced sampling and extensive augmentation. Such recipes motivate further experiments, but do not establish which changes would benefit Audio-JEPA. Our single-corpus, lightly augmented setup leaves these questions open..

7 Future Work

Four directions merit further investigation. First, the released predictor could be explored for detection based on masked latent-prediction error, potentially without spoofed training examples. This requires testing whether prediction errors distinguish genuine from spoofed speech and comparing latent-space scoring with input-space reconstruction error. Second, random, time-structured and frequency-structured masking during pre-training could be compared under otherwise matched conditions. Audio-JEPA uses random patch masking [30], whereas A-JEPA [10] anneals towards a SpecAugment-like scheme. Third, speech-focused pre-training could be compared with general-audio pre-training while controlling data scale and training budget. Fourth, downstream training with genuine speech from multiple corpora could test whether greater source diversity improves generalisation, using evaluation sources held out from model development.

8 Limitations

The comparison of Sect. 3.4 is between two released checkpoints, not between two objectives with all other factors held fixed. Both use a ViT-Base architecture and AudioSet-2M as their pre-training data source, and receive an identical downstream recipe. However, their pre-training recipes differ in masking ratio, sample rate, optimiser and compute

budget. Both encoders also receive inputs that differ from their pre-training configurations in distinct ways (Sect. 3.4); AudioMAE, for example, receives 256 rather than 1 024 frames. The effects of these input changes were not isolated. The proposed objective-related explanation in Sect. 6 therefore remains a hypothesis, not a demonstrated mechanism.

The evaluation sets used to inform configuration choices were not held out from model development. The system is trained on a single corpus, and three of the five evaluation corpora draw their genuine speech from other sources. Performance on these three shows limited generalisation under the present setup, but does not isolate source-corpus effects or establish the limits of the approach. Our complete system has 85.9 M parameters, compared with 318–340 M for the leading published systems in our comparison; we have not investigated how increasing encoder scale affects performance. Comparisons with published baseline systems also differ in back-end architecture and training recipe, and therefore assess complete systems rather than isolate encoder contributions.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y., Ballas, N.: Self-supervised learning from images with a joint-embedding predictive architecture. In: Proc. IEEE/CVF CVPR. pp. 15619–15629 (2023)
2. Babu, A., Wang, C., Tjandra, A., Lakhota, K., Xu, Q., Goyal, N., Singh, K., von Platen, P., Saraf, Y., Pino, J., Baevski, A., Conneau, A., Auli, M.: XLS-R: Self-supervised cross-lingual speech representation learning at scale. In: Proc. Interspeech. pp. 2278–2282 (2022)
3. Baevski, A., Zhou, Y., Mohamed, A., Auli, M.: wav2vec 2.0: A framework for self-supervised learning of speech representations. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 33 (2020)
4. Bardes, A., Garrido, Q., Ponce, J., Chen, X., Rabbat, M., LeCun, Y., Assran, M., Ballas, N.: Revisiting feature prediction for learning visual representations from video. In: Transactions on Machine Learning Research (2024)
5. Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., et al.: WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing* **16**(6), 1505–1518 (2022)
6. Desplanques, B., Thienpondt, J., Demuynck, K.: ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In: Proc. Interspeech. pp. 3830–3834 (2020)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Proc. ICLR (2021)
8. Dowerah, S., Kulkarni, A., Kulkarni, A., Tran, H.M., Kalda, J., Fedorchenko, A., Fauve, B., Lolive, D., Alumäe, T., Magimai Doss, M.: Speech DF arena: A leaderboard for speech deepfake detection models. arXiv:2509.02859v1 (2025)
9. Du, J., Lin, I.M., Chiu, I.H., Chen, X., Wu, H., Ren, W., Tsao, Y., Lee, H.y., Jang, J.S.R.: DFADD: The diffusion and flow-matching based audio deepfake dataset. In: Proc. IEEE Spoken Language Technology Workshop (SLT). pp. 921–928 (2024). <https://doi.org/10.1109/SLT61566.2024.10832250>, arXiv:2409.08731

10. Fei, Z., Fan, M., Huang, J.: A-JEPA: Joint-embedding predictive architecture can listen. arXiv:2311.15830 (2023)
11. Gemmeke, J.F., Ellis, D.P.W., Freedman, D., Jansen, A., Lawrence, W., Moore, R.C., Plakal, M., Ritter, M.: Audio set: An ontology and human-labeled dataset for audio events. In: Proc. IEEE ICASSP. pp. 776–780 (2017)
12. Hsu, W.N., Bolte, B., Tsai, Y.H.H., Lakhotia, K., Salakhutdinov, R., Mohamed, A.: HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Transactions on Audio, Speech, and Language Processing **29**, 3451–3460 (2021)
13. Huang, P.Y., Xu, H., Li, J., Baevski, A., Auli, M., Galuba, W., Metze, F., Feichtenhofer, C.: Masked autoencoders that listen. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 35 (2022)
14. Jung, J.w., Heo, H.S., Tak, H., Shim, H.j., Chung, J.S., Lee, B.J., Yu, H.J., Evans, N.: AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks. In: Proc. IEEE ICASSP. pp. 6367–6371 (2022)
15. Kawa, P., Plata, M., Czuba, M., Szymański, P., Syga, P.: Improved DeepFake detection using Whisper features. In: Proc. Interspeech. pp. 4009–4013 (2023)
16. LeCun, Y.: A path towards autonomous machine intelligence. OpenReview (2022), <https://openreview.net/forum?id=BZ5a1r-kVsf>, version 0.9.2, 27 June 2022
17. Lin, W., Wang, L., Wang, J., Feng, K., Wu, Z.: Tefic-Audio: Tell fact from fiction. arXiv:2607.28351v2 (2026)
18. Liu, T., Truong, D.T., Das, R.K., Lee, K.A., Li, H.: Nes2Net: A lightweight nested architecture for foundation model driven speech anti-spoofing. IEEE Transactions on Information Forensics and Security **20**, 12005–12018 (2025). <https://doi.org/10.1109/TIFS.2025.3626963>
19. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Proc. ICLR (2019)
20. Müller, N.M., Czempin, P., Dieckmann, F., Froghyar, A., Böttinger, K.: Does audio deepfake detection generalize? In: Proc. Interspeech. pp. 2783–2787 (2022)
21. Okabe, K., Koshinaka, T., Shinoda, K.: Attentive statistics pooling for deep speaker embedding. In: Proc. Interspeech. pp. 2252–2256 (2018)
22. Park, D.S., Chan, W., Zhang, Y., Chiu, C.C., Zoph, B., Cubuk, E.D., Le, Q.V.: SpecAugment: A simple data augmentation method for automatic speech recognition. In: Proc. Interspeech. pp. 2613–2617 (2019)
23. Speech Arena: Speech DF arena leaderboard. <https://huggingface.co/spaces/Speech-Arena-2025/Speech-DF-Arena> (2026), accessed 9 September 2026
24. Sun, C., Jia, S., Hou, S., Lyu, S.: AI-synthesized voice detection using neural vocoder artifacts. In: Proc. IEEE/CVF CVPR Workshops. pp. 904–912 (2023), arXiv:2304.13085
25. Tak, H., Jung, J.w., Patino, J., Kamble, M., Todisco, M., Evans, N.: End-to-end spectro-temporal graph attention networks for speaker verification anti-spoofing and speech deepfake detection. In: Proc. ASVspoof 2021 Workshop. pp. 1–8 (2021)
26. Tak, H., Patino, J., Todisco, M., Nautsch, A., Evans, N., Larcher, A.: End-to-end anti-spoofing with RawNet2. In: Proc. IEEE ICASSP. pp. 6369–6373 (2021)
27. Tak, H., Todisco, M., Wang, X., Jung, J.w., Yamagishi, J., Evans, N.: Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation. In: Proc. The Speaker and Language Recognition Workshop (Odyssey). pp. 112–119 (2022)
28. Todisco, M., Wang, X., Vestman, V., Sahidullah, M., Delgado, H., Nautsch, A., Yamagishi, J., Evans, N., Kinnunen, T., Lee, K.A.: ASVspoof 2019: Future horizons in spoofed and fake audio detection. In: Proc. Interspeech. pp. 1008–1012 (2019)
29. Truong, D.T., Tao, R., Nguyen, T., Luong, H.T., Lee, K.A., Chng, E.S.: Temporal-channel modeling in multi-head self-attention for synthetic speech detection. In: Proc. Interspeech (2024)

30. Tuncay, L., Labbé, E., Benetos, E., Pellegrini, T.: Audio-JEPA: Joint-embedding predictive architecture for audio representation learning. In: Proc. IEEE International Conference on Multimedia and Expo (ICME) Workshops (2025), arXiv:2507.02915
31. Wang, X., Yamagishi, J.: Investigating self-supervised front ends for speech spoofing countermeasures. In: Proc. The Speaker and Language Recognition Workshop (Odyssey). pp. 100–106 (2022)
32. Wang, X., Yamagishi, J., Todisco, M., Delgado, H., Nautsch, A., Evans, N., Sahidullah, M., Vestman, V., Kinnunen, T., Lee, K.A., et al.: ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech. *Computer Speech & Language* **64**, 101114 (2020)
33. Xiao, Y., Das, R.K.: XLSR-Mamba: A dual-column bidirectional state space model for spoofing attack detection. *IEEE Signal Processing Letters* **32**, 1276–1280 (2025). <https://doi.org/10.1109/LSP.2025.3547861>
34. Yamagishi, J., Veaux, C., MacDonald, K.: CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92). University of Edinburgh, The Centre for Speech Technology Research (CSTR), sound dataset (2019). <https://doi.org/10.7488/ds/2645>
35. Yang, Y., et al.: A robust audio deepfake detection system via multi-view feature. arXiv:2403.01960 (2024)
36. Yi, J., Fu, R., Tao, J., Nie, S., Ma, H., Wang, C., Wang, T., Tian, Z., Bai, Y., Fan, C., et al.: ADD 2022: The first audio deep synthesis detection challenge. In: Proc. IEEE ICASSP. pp. 9216–9220 (2022)
37. Zen, H., Dang, V., Clark, R., Zhang, Y., Weiss, R.J., Jia, Y., Chen, Z., Wu, Y.: LibriTTS: A corpus derived from LibriSpeech for text-to-speech. In: Proc. Interspeech. pp. 1526–1530 (2019)
38. Zhang, Q., Wen, S., Hu, T.: Audio deepfake detection with self-supervised XLS-R and SLS classifier. In: Proc. ACM International Conference on Multimedia (MM). pp. 6765–6773 (2024). <https://doi.org/10.1145/3664647.3681345>