LISTENING FORWARD: NEXT PATCH EMBEDDING PREDICTION ENABLES SCALABLE AUDIO LEARNERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-supervised learning has driven substantial progress in audio representation learning, though existing methods have increasingly relied on elaborate pre-training recipes to reach competitive performance. A markedly different pre-training philosophy underpins the most influential progress in language modeling and, more recently, in visual representation learning: rather than train encoders as static feature extractors, models are trained to predict the next element, a discrete token or a continuous embedding, from the preceding context. Autoregressive prediction thereby provides a unified pre-training interface that transfers across modalities, compelling the model to learn the underlying data distribution. We ask whether such a simple causal paradigm can yield strong audio learners, given that audio's temporal structure makes autoregressive prediction of patch embeddings a natural fit. We introduce NAPE (**N**ext-**A**udio-**P**atch-**E**mbedding prediction), a self-supervised framework in which a causal Transformer predicts each next patch embedding of a log-mel spectrogram from the previous ones, using causal masking and stop-gradient as its sole training signal. The design is intentionally minimalist, avoiding reconstruction decoders, acoustic tokenizers, student-teacher setups, and auxiliary regularization losses. Across six audio and speech benchmarks, NAPE achieves state-of-the-art fine-tuning performance and scales consistently across encoder sizes under both fine-tuning and linear probing. NAPE also produces structured attention patterns without explicit supervision.

1 INTRODUCTION

Self-supervised learning (SSL) has emerged as a foundational paradigm for representation learning across modalities (Radford et al., 2018; Devlin et al., 2019; Brown et al., 2020; Chen & He, 2021; He et al., 2022; Oquab et al., 2023; Baevski et al., 2020; Kong et al., 2024; Chen et al., 2024). In audio, SSL methods largely adapt paradigms first developed in vision: Vision Transformer backbones (Dosovitskiy et al., 2020) are pre-trained on spectrograms via *masked-spectrogram modeling* (Niizumi et al., 2022; Huang et al., 2022; Chong et al., 2023; Chen et al., 2022c; Dinkel et al., 2024; Niizumi et al., 2026) or *student-teacher distillation* objectives (Ahmed et al., 2024; Chen et al., 2024; Alex et al., 2025; Yang et al., 2026), typically pre-trained on AudioSet (Gemmeke et al., 2017) and evaluated by fine-tuning on downstream classification tasks.

These paradigms, however, rely on increasingly elaborate pre-training recipes: reconstruction decoders that map latent features back to raw mel content (Huang et al., 2022; Chen et al., 2024), separately-trained acoustic tokenizers (Chen et al., 2022c), teacher-student setups with exponential moving average (EMA)-updated encoders (Chen et al., 2024; Alex et al., 2025; Ahmed et al., 2024), auxiliary regularization losses (Fei et al., 2023; Alex et al., 2025), and multi-codebook vector quantisation (Yang et al., 2026). **Meanwhile, a distinct line of work in vision has shifted away from reconstruction toward the direct prediction of latent embeddings.** This extends a paradigm long established in language modeling, where models are trained not as static feature extractors but as predictive systems that model the data distribution through a single causal objective, providing a unified pre-training interface across modalities. Two variants have gained traction. *Joint-embedding predictive* approaches (Assran et al., 2023; Oquab et al., 2023; Fei et al., 2023; Bardes et al., 2024; Balestriero & LeCun, 2025; Yuksel et al., 2025; Huang et al., 2026; Wu et al., 2026) predict the latent embeddings of masked regions from a context view, but rely on heuristics (EMA, frozen layers, teacher-student architectures) for training stability. *Autoregressive next-embedding* approaches (Teoh

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.

et al., 2025; Xu et al., 2025; Bredis et al., 2026; Yao et al., 2026; Maes et al., 2026b), in contrast, predict each latent embedding directly from the preceding ones, mirroring the causal next-token objective of modern large language models (Radford et al., 2018; Brown et al., 2020; Liu et al., 2024).

Despite this progress, joint-embedding predictive approaches have seen *only limited exploration* in the audio and speech settings (Fei et al., 2023; Yuksel et al., 2025; Tuncay et al., 2025; Wang et al., 2026b), whilst predictive next-embedding are *entirely unexplored* in audio SSL. Their absence is particularly striking because audio is the modality *most* naturally suited to this paradigm: unlike images, whose 2D structure is approximately isotropic, audio signals unfold along a well-defined temporal axis, and their sequential structure is intrinsic to the signal rather than imposed on it. Predicting the next patch of a spectrogram from the preceding ones mirrors both how acoustic events emerge in time and how modern language models learn from sequential data.

Motivated by this alignment, we introduce NAPE (**N**ext-**A**udio-**P**atch-**E**mbedding prediction), a self-supervised framework that brings causal next-embedding prediction to audio. Given a log-mel spectrogram, NAPE first applies a patch embedding layer that produces a 2D grid of d-dimensional patch embeddings. Since the causal Transformer that follows operates on a 1D sequence, this grid must be linearized under a *scanning order*, a design axis specific to audio, since spectrograms have a strong temporal axis and a qualitatively different frequency axis, so the ordering determines both the causal context and the model's structural inductive bias. A causal Transformer encoder processes the linearized sequence, and a lightweight predictor head produces an estimate of the next patch embedding, analogous to next-token prediction in language modeling but operating in a continuous embedding space. Prediction quality is measured by negative cosine similarity to the target patch embedding under a stop-gradient: **the entire learning signal comes from the model's ability to anticipate the next embedding**, with no reconstruction decoder, acoustic tokenizer, student-teacher setup, or auxiliary regularization loss.

Extensive experiments on AudioSet (Gemmeke et al., 2017), ESC-50 (Piczak, 2015), Speech Commands V1 and V2 (Warden, 2018), and IEMOCAP (Busso et al., 2008) show that NAPE achieves state-of-the-art results, scales favorably across three encoder sizes (*Small* 19M, *Base* 85M, and *Large* 303M parameters), with linear-probing performance that improves consistently with encoder scale despite the purely predictive objective. A qualitative analysis further confirms that NAPE learns meaningful, structured features from audio. Our main contributions are:

- We introduce NAPE, the first self-supervised audio framework built around causal next-patch-embedding prediction, a substantially simpler alternative to reconstruction-based, masked-modeling, and joint-embedding approaches for audio.
- Through systematic ablations we identify the design axes that make next-embedding prediction work in the audio setting — scanning order, predictor head, prediction target, patch embedding layer, and the three key components of causality, prediction shift, and stop-gradient — and converge on an optimal configuration.
- Across six audio and speech benchmarks, NAPE delivers strong downstream results and exhibits favorable scaling behavior under both fine-tuning and linear-probing.
- We provide qualitative evidence that NAPE learns structured representations, with attention patterns that reason jointly about spectral context and frequency-consistent temporal history, and embeddings that group acoustically similar patches without any explicit supervision.

2 THE NAPE FRAMEWORK

In this section, we describe NAPE, a self-supervised pre-training method that trains a causal Transformer to predict the embedding of the next patch of a log-mel spectrogram from the preceding ones, relying only on causal masking and stop-gradient. Figure 1 depicts the overall framework.

2.1 SPECTROGRAM PATCHIFICATION AND SCANNING ORDER

Input Representation. Given a raw waveform, we compute a log-mel spectrogram $x \in \mathbb{R}^{1\times F\times T_{\text{frames}}}$ with F frequency bins and T_{frames} time frames. We split x into non-overlapping square patches of size $P \times P$, yielding a 2D grid of $N = T_F \cdot T_T$ patches with $T_F = F/P$

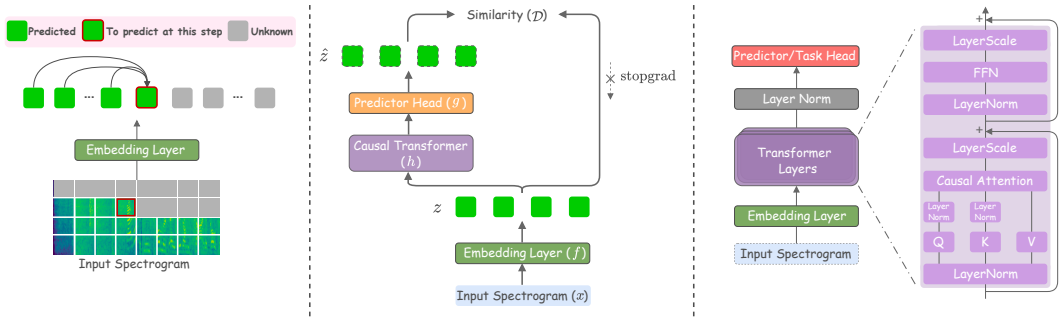


Figure 1: **Left:** NAPE's overview. The input spectrogram is split into patches and embedded into a sequence of embeddings. At each step, the model predicts the embedding of the next patch (red border) using only the embeddings of the preceding patches; patches at future positions are hidden by the causal attention mask. **Middle:** The NAPE pipeline: patch embeddings z are processed by the causal encoder h and predictor g to produce predictions $\hat{z}$, which are compared against the targets z under stop-gradient using a similarity function $\mathcal{D}$ (i.e., negative cosine similarity). **Right:** Encoder architecture (h): multiple stacked Transformer layers with pre-norm design, causal self-attention, LayerScale, and query-key normalization.

and $T_T = T_{\text{frames}}/P$. Each patch is projected to a d-dimensional embedding by a patch embedding layer f, producing the sequence $\{z_1, \ldots, z_N\} \in \mathbb{R}^{N \times d}$. We use a standard Conv2d embedding layer (Gong et al., 2021a; Huang et al., 2022) (we ablate this design choice in the Appendix C).

Scanning Order. Transformer encoders operate on 1D sequences, so a 2D patch grid must be linearized before it can be processed. Under bidirectional attention, as used in prior audio SSL methods (Gong et al., 2021a; Huang et al., 2022; Chen et al., 2024), the choice of ordering is not consequential: self-attention is permutation-equivariant given positional embeddings. NAPE's causal formulation breaks this equivariance: the ordering determines which patches are visible to a given position and which are predicted, imposing a substantive inductive bias, especially for spectrograms, whose temporal and frequency axes are qualitatively different, unlike the approximately isotropic structure of natural images. We consider four orderings, illustrated in Figure 2. ***Raster*** (left-to-right, bottom-to-top) sweeps time before advancing in frequency and is the standard patch ordering in vision and audio Transformer models. ***Time-major*** sweeps frequency before advancing in time, conditioning each prediction on the full frequency profile of every past time step. ***Zigzag*** is an alternating-direction variant of raster: rows alternate left-to-right and right-to-left, avoiding the spatial discontinuity raster incurs across row transitions. ***Diagonal*** sweeps patches along anti-diagonals of the grid, mixing time and frequency progression at every step.

2.2 Next-Audio-Patch-Embedding Prediction

Prediction Task. Given the embedding sequence $z = \{z_1, \ldots, z_N\}$ produced by f in the chosen scanning order, NAPE trains an encoder h and a lightweight predictor head g so that, at each position t, the model produces an estimate of the next patch embedding z_{t+1} using only the previous patches:

$$\hat{z}_{t+1} = g(h(z_{\leq t}))\,, \tag{1}$$

where $z_{\leq t}$ denotes the embeddings at all positions up to and including t. The restriction to the past patches is enforced by a *causal attention mask* that prevents each position from attending to patches at later positions in the sequence. This is directly analogous to next-token prediction in language modeling, but operates in a continuous embedding space rather than over a discrete vocabulary.

Loss Function. Following SimSiam (Chen & He, 2021), we measure prediction quality by the negative cosine similarity between the target embedding z_{t+1} and the predicted embedding $\hat{z}_{t+1}$:

$$\mathcal{D}(z_{t+1}, \hat{z}_{t+1}) \;=\; -\,\frac{z_{t+1}}{\|z_{t+1}\|_2} \cdot \frac{\hat{z}_{t+1}}{\|\hat{z}_{t+1}\|_2}, \tag{2}$$

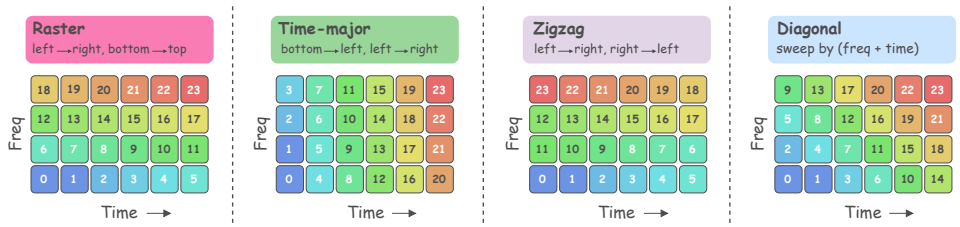


Figure 2: **Patch scanning orders.** NAPE linearizes the 2D spectrogram patch grid into a 1D causal sequence in one of four ways: *raster* (left-to-right, bottom-to-top), *time-major* (bottom-to-top within each time column, then advance in time), *zigzag* (raster with alternating row directions), and *diagonal* (sweep by frequency-plus-time index). The numbers indicate each patch position in the sequence.

where $\|\cdot\|_2$ is the ℓ_2-norm. Applying stop-gradient (`stopgrad`) on the target and averaging over all valid prediction positions yields the NAPE objective:

$$\mathcal{L} \;=\; \frac{1}{N-1}\sum_{t=1}^{N-1} \mathcal{D}(\texttt{stopgrad}(z_{t+1}),\; \hat{z}_{t+1})\,. \tag{3}$$

The `stopgrad` operator treats the target embedding as a constant, so gradients flow only through the predicted side. Cosine similarity is magnitude-invariant, which prevents the trivial solution of shrinking both sides of the objective toward zero norm (we ablate the choice of $\mathcal{D}$ in the Appendix C).

Causal Transformer Encoder (h). We use a Vision Transformer backbone (Dosovitskiy et al., 2020; Wang et al., 2026a) with pre-norm design and a causal attention mask during pre-training (see Figure 1, right panel). For stability at depth, we adopt Rotary Position Embedding (RoPE) (Su et al., 2024) applied independently along the frequency and time axes, LayerScale (Touvron et al., 2021) on residual branches, and parameter-free query-key normalization (Henry et al., 2020) on the per-head query and key projections. We instantiate three configurations of increasing size to test NAPE's scalability: *Small* ($d = 512$, 6 layers, 8 heads; $\sim$19M parameters), *Base* ($d = 768$, 12 layers, 12 heads; $\sim$85M parameters), and *Large* ($d = 1024$, 24 layers, 16 heads; $\sim$303M parameters).

Predictor Head (g). The predictor head g decouples the representation space of h from the space in which predictions are made, avoiding the constraint that the encoder place its representations directly in the target space. This use of an asymmetric predictor follows SimSiam (Chen & He, 2021), though the two settings differ: SimSiam compares two augmented views produced by a symmetric encoder, whereas NAPE's branches are asymmetric by construction, comparing a prediction produced by the deep encoder h at position t against the shallow embedding of the actual next patch produced by f. We ablate multiple predictor styles in Section 3.2.

3 Experiments

3.1 Experimental Setup

Pre-training Data. We pre-train NAPE on AudioSet (Gemmeke et al., 2017) without labels, combining the unbalanced and balanced training splits. We obtained and processed 1,964,222 clips from the unbalanced split, 20,961 clips from the balanced split, and 18,900 clips from the evaluation split, consistent with prior work (Chen et al., 2022c; 2024; Alex et al., 2025). All input waveforms are resampled to mono at 16 kHz and converted into log-mel spectrograms with 128 mel bands using a 25 ms Hanning window and 10 ms hop size. A 10-second clip yields a spectrogram of size $1 \times 128 \times 1008$ (channel, frequency, time), corresponding to $8 \times 63 = 504$ non-overlapping 16×16 patches per clip (as in (Gong et al., 2021a; Chen et al., 2022c; 2024)).

Downstream Benchmarks. Following prior work, for downstream evaluation, we fine-tune NAPE on **AudioSet-2M** (*AS-2M*, unbalanced) and **AudioSet-20K** (*AS-20K*, balanced), applying the weighted sampling strategy of Huang et al. (2022) on AS-2M to mitigate class imbalance. We further evaluate on **ESC-50** (Piczak, 2015), a 50-class environmental sound classification benchmark of 2000 clips at

5s each, using 5-fold cross-validation; **Speech Commands V1 and V2** (*KS1*, *KS2*) (Warden, 2018), keyword spotting tasks with 12 and 35 classes respectively; and **IEMOCAP** (*ER*) (Busso et al., 2008), a 4-class speech emotion recognition benchmark with 5-fold cross-validation. Together these tasks span both audio and speech domains. We also discuss in Appendix J how NAPE could be extended to speech pre-training and to multimodal speech settings. We report mean average precision (mAP) on the multi-label AudioSet tasks and top-1 classification accuracy on the single-label tasks; for ESC-50 and IEMOCAP we report the mean across cross-validation folds.

Pre-training Details. We pre-train with the AdamW optimizer (Loshchilov & Hutter, 2017) using a base learning rate of 5×10^{-3} with cosine decay, a weight decay of 0.05, $\beta_1 = 0.9$, $\beta_2 = 0.95$, and a batch size of $256/256/128$ for NAPE Small/Base/Large. We pre-train NAPE Small and Large for 25 epochs and NAPE Base for 30 epochs, with a warmup ratio of 10%. All models are trained using HuggingFace `Trainer` with distributed data parallelism (DDP) on NVIDIA L40s GPUs.

Fine-tuning Details. For each downstream task, we initialize from the pre-trained model (*the predictor is discarded after pre-training*) and attach a linear classifier on top of mean-pooled patch tokens. We disable the causal attention mask during fine-tuning so that attention is bidirectional over the full patch sequence. We provide an ablation study on the optimal pooling method and whether to use causal/bidirectional attention in the Appendix C. We use AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, cosine learning-rate decay, and layer-wise learning-rate decay (Clark et al., 2020; Bao et al., 2021). We use binary cross-entropy for AudioSet and soft-target cross-entropy for the single-label tasks. Following prior work (Gong et al., 2021a; Chen et al., 2024; 2022c), we apply a standard augmentation stack (SpecAugment (Park et al., 2019), Mixup (Zhang et al., 2017), CutMix (Yun et al., 2019), DropPath (Huang et al., 2016), temporal roll, additive noise, and label smoothing (Szegedy et al., 2016)). For AS-2M and AS-20K we additionally maintain an exponential moving average (EMA) (Polyak & Juditsky, 1992) of the fine-tuning weights (decay 0.99995 and 0.999 respectively); no EMA is used for the other tasks. Full task-specific hyperparameters are reported in the Appendix B.

Linear Probing Details. For linear probing, we freeze the pre-trained model and train only a linear classifier. All augmentations are disabled.

Baselines. We compare against recent in-domain self-supervised methods, including Audio-MAE (Huang et al., 2022), BEATs (Chen et al., 2022c), A-JEPA (Fei et al., 2023), ASiT (Ahmed et al., 2024), EAT (Chen et al., 2024), SSAMBA (Shams et al., 2024), SSLAM (Alex et al., 2025), and SPEAR (Yang et al., 2026). We also include results from out-of-domain and in-domain supervised pre-training. For all baseline methods, fine-tuning results are taken from the original papers.

3.2 NAPE'S OPTIMAL CONFIGURATION

In this section, we ablate some of the key design axes of NAPE to identify its optimal configuration. Unless otherwise specified, all ablations use the NAPE *Base* model and the raster scanning order. Each ablation modifies a single design axis at a time, holding the rest fixed at NAPE's defaults.

Prediction Shift/Stop-gradient/Causality. Table 1 disentangles three key mechanisms of NAPE. Without the *prediction shift*, the objective reduces to an identity mapping that provides no meaningful target, and the resulting checkpoint fails to converge to a usable model during fine-tuning. Without the *stop-gradient*, the training loss collapses immediately to -1 during pre-training as the encoder drives all embeddings toward a shared constant, the representation collapse that the stop-gradient exists to prevent. Removing the *causal mask* degrades downstream performance sharply, with the largest drops on the audio benchmarks and smaller but consistent drops on speech tasks): with bidirectional attention the encoder can attend to the target patch while producing its prediction, so the objective is satisfied by a near-identity mapping that routes each target back to its predicted position.

Table 1: **Ablation on main NAPE's design elements**: *prediction shift*, *stop-gradient*, and *causal objective*.

pred shift	**stop grad**	**causal mask**	**Audio Tasks** *AS-2M*	*AS-20K*	*ESC-50*	**Speech Tasks** *KS1*	*KS2*	*ER*
✘	✓	✓			**Fail**			
✓	✘	✓			**Fail**			
✓	✓	✘	41.8	24.8	68.9	96.1	97.3	57.0
✓	✓	✓	**49.6**	**39.1**	**94.2**	**97.9**	**98.8**	**64.9**

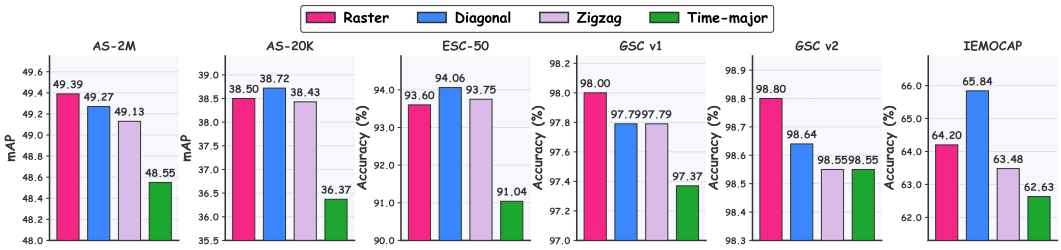


Figure 3: NAPE's performance across four scan orders on six benchmarks.

Predictor g. Table 2 compares four predictor variants: no predictor (encoder output used directly as the prediction), a two-layer MLP (*2-MLP*; Linear$(d, d) \rightarrow$ GELU $\rightarrow$ Linear(d, d)), a *SimSiam*-style (Chen & He, 2021) three-layer predictor with intermediate LayerNorms, and a *Transformer* predictor in the style of JEPA-family methods (a 2-layer causal Transformer with 16 heads). Adding any predictor improves over using the encoder output directly, confirming that decoupling the representation and prediction spaces benefits the learned features. The SimSiam variant performs best on the audio benchmarks and IEMOCAP, and is on par on the keyword-spotting tasks. Notably, it outperforms the Transformer predictor despite being nearly $8\times$ smaller, consistent with the finding in the siamese self-supervised setting (Chen & He, 2021) that a compact MLP predictor suffices with a well-designed encoder and stop-gradient target. The SimSiam-style predictor is thus NAPE's default.

Table 2: Ablation on the predictor-style variants.

Predictor Style	#Par.	Audio Tasks			Speech Tasks		
		AS-2M	*AS-20K*	*ESC-50*	*KS1*	*KS2*	*ER*
None	-	48.7	37.8	93.3	98.1	**98.8**	64.2
2-MLP	1.2M	49.4	38.5	93.6	98.0	**98.8**	64.2
Transformer	14.2M	49.2	38.4	93.0	**98.2**	98.7	**65.0**
SimSiam	1.8M	**49.6**	**39.1**	**94.2**	97.9	**98.8**	64.9

Patch Scanning Variants. Figure 3 compares the four scanning orders introduced in Section 2.1 (we use the 2-MLP predictor style). *Diagonal*, *raster*, and *zigzag* all perform comparably well across the six benchmarks, with diagonal and raster slightly ahead of zigzag on most tasks. In contrast, *time-major* consistently underperforms the other three orders across all tasks. We interpret this pattern as reflecting the temporal structure of audio: diagonal, raster, and zigzag all advance in time as the causal sequence progresses, allowing the model to accumulate temporal context in a way that matches how acoustic events unfold. Time-major, instead, exhausts each frequency column before advancing in time, so predictions early in the sequence are conditioned on rich instantaneous spectra but only limited temporal context, a mismatch with the temporal nature of audio classification. Given their superior performance, we retain both raster and diagonal in the next sections.

Prediction Target. Table 3 compares three targets for the autoregressive prediction: (i) the *patch embedding* z_{t+1} produced by f (the default); (ii) the *raw mel* content of the next patch, following masked reconstruction methods such as Audio-MAE (Huang et al., 2022); (iii) the output of the *first encoder layer*, in the style of JEPA methods that use deeper encoder features as prediction targets. For the raw mel variant, we add a linear projection on top of g to map its output to the raw patch dimensionality. Patch embedding and raw mel yield comparable results across all benchmarks. Using the first encoder layer, however, causes pre-training to collapse: the target itself depends on the encoder being trained, and stop-gradient alone cannot prevent both sides of the loss from collapsing to a shared constant. JEPA-family methods circumvent this instability with additional regularizations (EMA teachers (Assran et al., 2023; Fei et al., 2023), or sketching-based regularizers such as SIGReg (Balestriero & LeCun, 2025) and VISReg (Wu et al., 2026))), which we do not employ.

Table 3: Ablation on NAPE's target to predict.

Predicted Target	Audio Tasks			Speech Tasks		
	AS-2M	*AS-20K*	*ESC-50*	*KS1*	*KS2*	*ER*
1st enc. layer			**Fail**			
Raw mel	**49.7**	38.0	**94.8**	97.7	98.6	64.2
Patch embed	49.6	**39.1**	94.2	**97.9**	**98.8**	**64.9**

Additional Ablation Results. We refer to Appendix C for additional ablations studies on: **1)** the similarity function $\mathcal{D}$, **2)** the patch embedding layer f, **3)** the use of normalization layers (LayerNorm vs RMSNorm), **4)** freezing/unfreezing the patch embedding layer, **5)** the optimal positional encoding

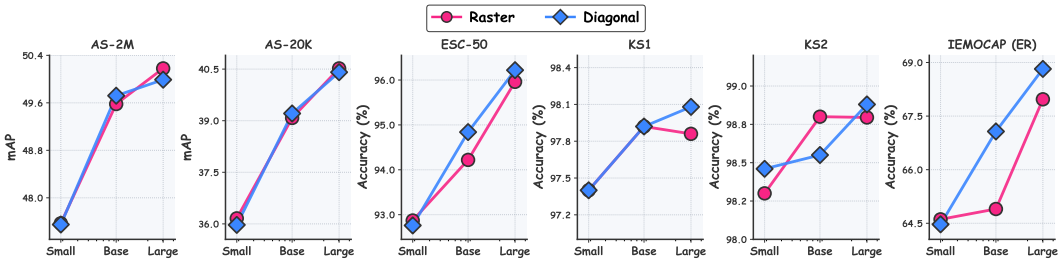


Figure 5: NAPE's results at different scales under raster and diagonal scan orders.

(absolute encoding vs RoPE), **6)** the optimal pooling method during fine-tuning, **7)** the optimal attention type during fine-tuning (causal vs bidirectional), **(8)** the additional use of random masking during pre-training, and **9)** NAPE's performance in terms of different compute budgets.

NAPE's Optimal Configuration

(i) **Conv2d** embedding layer, (ii) **raster/diagonal** scanning order, (iii) **SimSiam-style predictor**, (iv) **patch embedding** as the prediction target, and (v) **negative cosine similarity** loss with stop-gradient on the target branch.

3.3 Scaling NAPE

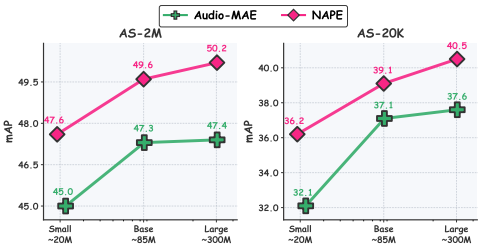


Figure 4: Scaling comparison between Audio-MAE and NAPE, raster.

Figure 5 shows downstream performance for the raster and diagonal variants of NAPE at three encoder scales: *Small* (∼19M parameters, NAPE-S), *Base* (∼85M, NAPE-B), and *Large* (∼303M, NAPE-L). NAPE scales positively across the board: on 10 out of 12 configurations, NAPE-L improves over NAPE-B, which in turn always improves over the small version. While this improvement tends to diminish as we scale from the base to the large model, we observe consistent scaling gains for most of the tasks, while the keyword-spotting tasks show smaller absolute gains, reflecting their already-saturated accuracy levels. Raster and diagonal track each other closely at the Small scale, with essentially identical performance across all six tasks. At the Base and Large scales, diagonal slightly outperforms raster on most benchmarks, while raster achieves the strongest result on AS-2M with NAPE-L reaching 50.18 mAP. In Figure 4, we compares NAPE (raster) and Audio-MAE (Huang et al., 2022) at three encoder scales on AS-2M and AS-20K. NAPE outperforms Audio-MAE at every scale, with particularly large margins at the Small size (+2.6 mAP on AS-2M, +4.1 mAP on AS-20K). NAPE also exhibits more favorable scaling behavior from Base to Large than Audio-MAE.

3.4 Comparison with State-of-the-Art

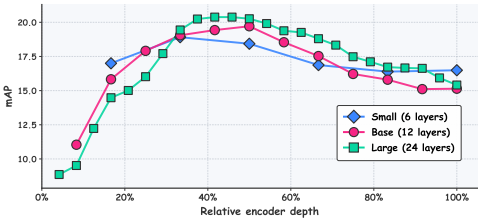


Figure 6: Layer-wise linear probing analysis.

Table 4 compares NAPE against prior audio pre-training methods. At the Base scale, NAPE-B diagonal obtains the strongest AS-2M result (40.7 mAP) among comparably sized self-supervised methods other than SSLAM, and the strongest IEMOCAP accuracy, while remaining competitive on AS-20K and on the keyword-spotting tasks. At the Large scale, NAPE exceeds the size-matched SPEAR Large on both AudioSet benchmarks and is on par with the strongest baseline, SSLAM, tying it on AS-2M (50.2 mAP) and on ESC-50 (96.2%), and approaching it on AS-20K. What is notable is the simplicity of NAPE's recipe: unlike SSLAM, which combines audio-mixture supervision, a student-teacher architecture, and a reconstruction decoder, NAPE requires only a stop-gradient and negative cosine similarity between the predicted next patch embedding and its target. This simplicity is also reflected in *cost*: NAPE-B is the **least expensive to pre-train** among the baselines we compare against (see Appendix E). NAPE also outperforms A-JEPA on four of the five benchmarks it reports, despite

Table 4: Comparison with audio methods on audio and speech downstream tasks. IN, AS, and LS denote the ImageNet, AudioSet, and LibriSpeech datasets, respectively. TI denotes the 400M text-image pairs for CLIP pre-training. We **gray out** the models and results with additional supervised training on external datasets. Best results are in **bold**, second-best are underlined.

Model	#Par.	Pre-train Data	Audio Tasks			Speech Tasks		
			AS-2M	*AS-20K*	*ESC-50*	*KS1*	*KS2*	*ER*
Out-of-domain Supervised Pre-training								
PSLA (Gong et al., 2021b)	14M	IN	44.4	31.9	-	-	96.3	-
AST (Gong et al., 2021a)	86M	IN	45.9	34.7	88.7	95.5	98.1	56.0
HTS-AT (Chen et al., 2022a)	31M	IN	47.1	-	-	-	98.0	-
Audio-CLIP (Guzhov et al., 2022)	93M	TI+AS	25.9	-	96.7	-	-	-
In-domain Supervised Pre-training								
AST (Gong et al., 2021a)	86M	IN+AS	45.9	-	95.6	-	97.9	-
HTS-AT (Chen et al., 2022a)	31M	IN+AS	47.1	-	97.0	-	-	-
Audio-MAE (Huang et al., 2022)	86M	AS	-	-	97.4	-	-	-
Self-Supervised Pre-training								
SS-AST (Gong et al., 2022a)	89M	AS+LS	-	31.0	88.8	96.0	98.0	59.6
MAE-AST (Baade et al., 2022)	86M	AS+LS	-	30.6	90.0	95.8	97.9	59.8
CAV-MAE (Gong et al., 2022b)	86M	IN+AS	44.9	34.2	-	-	-	-
Audio-MAE (Huang et al., 2022)	86M	AS	47.3	37.1	94.1	96.9	98.3	-
Audio-MAE L (Huang et al., 2022)	304M	AS	47.4	37.6	-	-	-	-
data2vec (Baevski et al., 2022)	94M	AS	-	34.5	-	-	-	-
MaskSpec (Chong et al., 2023)	86M	AS	47.1	32.3	89.6	-	97.7	-
BEATs$_{iter3}$ (Chen et al., 2022c)	90M	AS	48.0	38.3	95.6	97.7	98.3	64.5
A-JEPA (Fei et al., 2023)	86M	AS	48.6	38.4	**96.3**	97.7	98.5	-
ASiT (Ahmed et al., 2024)	86M	AS	48.0	38.6	95.3	98.2	**98.9**	-
EAT (Chen et al., 2024)	88M	AS	48.6	40.2	95.9	-	98.3	-
SSAMBA (Shams et al., 2024)	99M	AS+LS	-	28.3	89.3	96.9	97.4	61.5
SSLAM (Alex et al., 2025)	88M	AS	**50.2**	**40.9**	96.2	**98.8**	98.1	-
SPEAR$_a$ Large (Yang et al., 2026)	327M	AS	49.7	39.3	-	-	-	-
NAPE-B raster	85M	AS	49.6	39.1	94.2	97.9	98.8	64.9
NAPE-B diagonal	85M	AS	49.7	39.2	94.8	97.9	98.6	67.1
NAPE-L raster	303M	AS	**50.2**	40.5	96.0	97.9	98.8	68.0
NAPE-L diagonal	303M	AS	50.0	40.4	96.2	98.2	**98.9**	**68.8**

A-JEPA using an EMA target encoder, parallel masked prediction, and a regularized masking strategy at fine-tuning. Finally, NAPE transfers strongly to speech tasks: NAPE-L diagonal reaches an accuracy of 98.2, 98.9, and 68.8 on KS1, KS2 and IEMOCAP, respectively.

3.5 Linear Probing Results

Which Layer to Probe? Linear probing evaluates the separability of the pre-trained features by freezing the encoder and training only a linear classification head. Following recent observations that classification-relevant features often lie in intermediate rather than final layers (Skean et al., 2025; Bolya et al., 2026), we first study downstream performance as a function of the encoder depth. Figure 6 reports AS-20K mAP when probing each layer of NAPE-S, NAPE-B, and NAPE-L. Across all three scales, the best probing layer lies at roughly the middle of the encoder (layer 2, 6, and 11 respectively), with performance dropping as we approach the initial or final layers.

Table 5: Linear probing results.

Model	Layer	*AS-2M*	*AS-20K*	*ESC-50*
Small	2nd	23.2	18.9	79.8
Base	6th	25.0	19.7	81.7
Large	11th	**27.1**	**20.4**	**83.5**

Linear Probing Across Scales and Tasks. Using the best probing layer per model, we compare NAPE-S, NAPE-B, and NAPE-L on AS-2M, AS-20K, and ESC-50. Table 5 reports the results. NAPE scales positively under linear probing: larger models yield stronger probes on every task, reinforcing the scaling behavior observed under fine-tuning (Section 3.3). Absolute linear-probing scores remain below their fine-tuning counterparts, as expected for prediction-based SSL methods since the next-patch prediction objective does not directly optimize for linear separability.

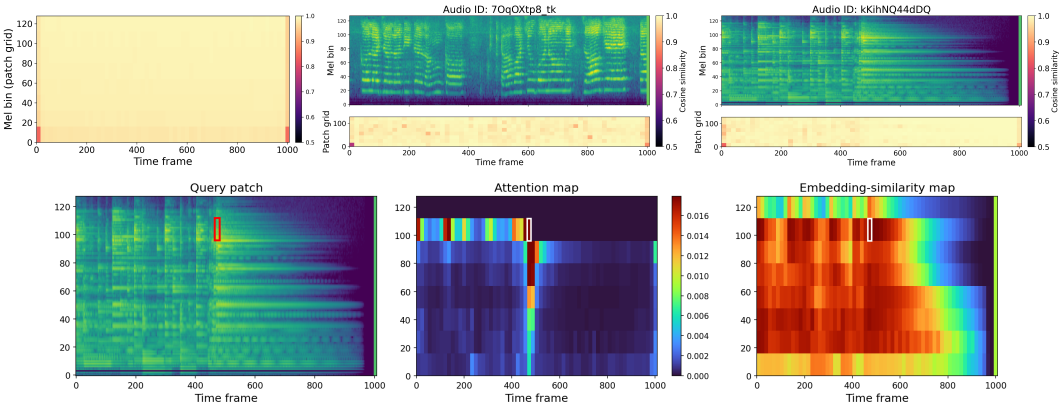


Figure 7: **Top:** Prediction Quality Analysis. We report the cosine similarity between the predicted and true patch embeddings, averaged over 500 audio clips (**left**) and for two individual clips (**middle**, **right**). **Bottom:** Attention/Embedding Analyses. **Left:** selected query patch; **middle:** attention map showing the patches NAPE attends to when predicting the next patch; **right:** embedding-similarity map between the predicted next-patch embedding $\hat{z}_{t+1}$ and every actual patch embedding.

3.6 Qualitative Results

To gain insight into what NAPE learns, we complement the quantitative results with two qualitative analyses of a pre-trained NAPE-L (raster) on the AudioSet evaluation set. Both use the model after pre-training, with no fine-tuning. More qualitative results can be found in the Appendix H.

Prediction Fidelity. We measure the cosine similarity between the predicted embeddings $\hat{z}_{t+1}$ and the true patch embeddings z_{t+1}. Figure 7 (top left) shows the similarity averaged across 500 held-out AudioSet clips, and the top middle and right panels report the same measurement for two individual clips: NAPE predicts the next patch embedding accurately almost everywhere on the grid, with similarity close to the ceiling of 1.0 both on average and per clip. The remaining low-similarity regions have clear structural explanations. The very first patch has the lowest similarity, since no previous context is available for the prediction to condition on. The first mel row is harder on average, since it combines limited past context with low-frequency patches. Finally, the rightmost patches show slightly lower similarity because the last time columns correspond to zero-padded frames.

Attention and Embedding Analysis. To understand *how* NAPE arrives at its predictions, we select a query patch and analyze the attention and embedding structure it induces. The attention map (Figure 7, bottom middle), which conveys the aggregated attention from the query position to every other patch averaged over all layers and heads, reveals a highly structured pattern with two distinct components. NAPE attends strongly to the current time column, integrating the full spectral profile of the current moment, and to same-mel-frequency patches earlier in the clip, tracking the temporal evolution of the frequency it is about to predict. The embedding-similarity map (bottom right), which compares the predicted embedding $\hat{z}_{t+1}$ against every actual patch embedding in the clip, shows that $\hat{z}_{t+1}$ is most similar to patches that share acoustic structure and energy with the query, regions of the spectrogram carrying comparable spectral content. *This grouping into coherent acoustic components emerges without any explicit labels or region annotations.*

4 Conclusion

We presented NAPE, a self-supervised framework for audio representation learning based on causal next patch embedding prediction. Departing from the reconstruction- and masking-based approaches that dominate audio SSL, NAPE relies on a deliberately minimalist recipe: a single causal Transformer encoder, a lightweight predictor head, and a negative cosine similarity loss with stop-gradient on the target branch. Across six audio and speech benchmarks, NAPE achieves strong downstream performance and exhibits favorable scaling behavior across three encoder sizes for both fine-tuning and linear probing. NAPE also develops structured attention patterns and organizes its learned embeddings into acoustically coherent regions without any explicit supervision.

AI USE STATEMENT

We have not used generative AI tools for any of the tasks for which ICLR requires disclosure. Additionally, we used generative AI tools for two tasks with recommended disclosure: editing the manuscript to improve readability, and auxiliary review of parts of the software code. Editing was restricted to grammar, phrasing and concision on text first written by the authors; no section of the paper was drafted by a generative AI tool, and all claims, numbers and citations are the authors' own. Code assistance was limited to reviewing selected components of the training and evaluation pipeline for correctness. All flagged issues were inspected by the authors, and every reported result was produced by code that the authors verified and executed themselves. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have taken several steps to ensure that our work is reproducible. The NAPE pre-training and downstream fine-tuning recipes are described in full in Section 3.1, including model configurations, optimizer settings, and training schedules. All ablation configurations reported in Section 3.2 follow the shared setup described at the start of that section, with only the specific design axis under study varying. The full pseudocode of the NAPE pre-training loop is provided in Algorithm 1, and additional hyperparameter details for each downstream benchmark are reported in the Appendix B. We rely exclusively on publicly available datasets (AudioSet, ESC-50, Speech Commands V1/V2, and IEMOCAP) and describe our data processing pipeline in Section 3.1. Upon acceptance, we will release the full source code for pre-training, fine-tuning, linear probing, and the qualitative analyses, together with pre-trained NAPE checkpoints at all three scales, so that the results reported in this paper can be fully reproduced end-to-end.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Sara Atito Ali Ahmed, Muhammad Awais, Wenwu Wang, Mark D Plumbley, and Josef Kittler. Asit: Local-global audio spectrogram vision transformer for event classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:3684–3693, 2024.

Tony Alex, Sara Atito, Armin Mustafa, Muhammad Awais, and Philip Jackson. Sslam: Enhancing self-supervised models with audio mixtures for polyphonic soundscapes. In *International Conference on Learning Representations*, volume 2025, pp. 22608–22626, 2025.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Alan Baade, Puyuan Peng, and David Harwath. Mae-ast: Masked autoencoding audio spectrogram transformer. *arXiv preprint arXiv:2203.16691*, 2022.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International conference on machine learning*, pp. 1298–1312. PMLR, 2022.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Bangalath, et al. Perception encoder: The best visual embeddings are not at the output of the network. *Advances in Neural Information Processing Systems*, 38:60884–60937, 2026.

George Bredis, Nikita Balagansky, Daniil Gavrilov, and Ruslan Rakhimov. Next embedding prediction makes world models stronger. *arXiv preprint arXiv:2603.02765*, 2026.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335–359, 2008.

Umberto Cappellazzo, Minsu Kim, Honglie Chen, Pingchuan Ma, Stavros Petridis, Daniele Falavigna, Alessio Brutti, and Maja Pantic. Large language models are strong audio-visual speech recognition learners. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Umberto Cappellazzo, Minsu Kim, Pingchuan Ma, Honglie Chen, Xubo Liu, Stavros Petridis, and Maja Pantic. Mome: Mixture of matryoshka experts for audio-visual speech recognition. *Advances in neural information processing systems*, 38:40013–40040, 2026a.

Umberto Cappellazzo, Xubo Liu, Pingchuan Ma, Stavros Petridis, and Maja Pantic. Omni-avsr: Towards unified multimodal speech recognition with large language models. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 17772–17776. IEEE, 2026b.

Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 646–650. IEEE, 2022a.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pp. 1691–1703. PMLR, 2020.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, 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, 2022b.

Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, and Furu Wei. Beats: Audio pre-training with acoustic tokenizers. *arXiv preprint arXiv:2212.09058*, 2022c.

Wenxi Chen, Yuzhe Liang, Ziyang Ma, Zhisheng Zheng, and Xie Chen. Eat: Self-supervised pre-training with efficient audio transformer. *arXiv preprint arXiv:2401.03497*, 2024.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 15745–15753. IEEE, 2021.

Dading Chong, Helin Wang, Peilin Zhou, and Qingcheng Zeng. Masked spectrogram prediction for self-supervised audio pre-training. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Heinrich Dinkel, Zhiyong Yan, Yongqing Wang, Junbo Zhang, Yujun Wang, and Bin Wang. Scaling up masked audio encoder learning for general audio classification. *arXiv preprint arXiv:2406.06992*, 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap learning audio concepts from natural language supervision. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Zhengcong Fei, Mingyuan Fan, and Junshi Huang. A-jepa: Joint-embedding predictive architecture can listen. *arXiv preprint arXiv:2311.15830*, 2023.

Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 776–780. IEEE, 2017.

Houtan Ghaffari, Lukas Rauch, Christoph Scholz, and Paul Devos. Bat: Better audio transformer guided by convex gated probing. In *International conference on machine learning*, 2026.

Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer. In *INTERSPEECH*, 2021a.

Yuan Gong, Yu-An Chung, and James Glass. Psla: Improving audio tagging with pretraining, sampling, labeling, and aggregation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3292–3306, 2021b.

Yuan Gong, Cheng-I Lai, Yu-An Chung, and James Glass. Ssast: Self-supervised audio spectrogram transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 10699–10709, 2022a.

Yuan Gong, Andrew Rouditchenko, Alexander H Liu, David Harwath, Leonid Karlinsky, Hilde Kuehne, and James Glass. Contrastive audio-visual masked autoencoder. *arXiv preprint arXiv:2210.07839*, 2022b.

Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. Audioclip: Extending clip to image, text and audio. In *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 976–980. IEEE, 2022.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 15979–15988. IEEE, 2022.

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4246–4253, 2020.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 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.

Chen Huang, Xianhang Li, Vimal Thilak, Etai Littwin, and Josh Susskind. Text-conditional jepa for learning semantically rich visual representations. *arXiv preprint arXiv:2605.03245*, 2026.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pp. 646–661. Springer, 2016.

Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. Masked autoencoders that listen. *Advances in neural information processing systems*, 35:28708–28720, 2022.

Zhifeng Kong, Arushi Goel, Rohan Badlani, Wei Ping, Rafael Valle, and Bryan Catanzaro. Audio flamingo: A novel audio language model with few-shot learning and dialogue abilities. *arXiv preprint arXiv:2402.01831*, 2024.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Pingchuan Ma, Alexandros Haliassos, Adriana Fernandez-Lopez, Honglie Chen, Stavros Petridis, and Maja Pantic. Auto-avsr: Audio-visual speech recognition with automatic labels. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Lucas Maes, Quentin Le Lidec, Luiz Facury, Nassim Massaudi, Ayush Chaurasia, Francesco Capuano, Richard Gao, Taj Gillin, Dan Haramati, Damien Scieur, et al. stable-worldmodel: A platform for reproducible world modeling research and evaluation. *arXiv preprint arXiv:2605.21800*, 2026a.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026b.

Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Noboru Harada, and Kunio Kashino. Masked spectrogram modeling using masked autoencoders for learning general-purpose audio representation. In *HEAR: Holistic Evaluation of Audio Representations*, pp. 1–24. PMLR, 2022.

Daisuke Niizumi, Daiki Takeuchi, Masahiro Yasuda, Binh Thien Nguyen, Noboru Harada, and Nobutaka Ono. Rethinking masking strategies for masked prediction-based audio self-supervised learning. *arXiv preprint arXiv:2603.23810*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

Karol J Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 1015–1018, 2015.

Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *OpenAI Tech Report*, 2018.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763, 2021.

Lukas Rauch, René Heinrich, Houtan Ghaffari, Lukas Miklautz, Ilyass Moummad, Bernhard Sick, and Christoph Scholz. Unmute the patch tokens: Rethinking probing in multi-label audio classification. In *International Conference on Learning Representations*, volume 2026, pp. 62861–62883, 2026.

Siavash Shams, Sukru Samet Dindar, Xilin Jiang, and Nima Mesgarani. Ssamba: Self-supervised audio representation learning with mamba state space model. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pp. 1053–1059. IEEE, 2024.

Bowen Shi, Wei-Ning Hsu, Kushal Lakhotia, and Abdelrahman Mohamed. Learning audio-visual speech representation by masked multimodal cluster prediction. *arXiv preprint arXiv:2201.02184*, 2022.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Jayden Teoh, Manan Tomar, Kwangjun Ahn, Edward S Hu, Tim Pearce, Pratyusha Sharma, Akshay Krishnamurthy, Riashat Islam, Alex Lamb, and John Langford. Next-latent prediction transformers learn compact world models. *arXiv preprint arXiv:2511.05963*, 2025.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 32–42. IEEE, 2021.

Ludovic Tuncay, Etienne Labbé, Emmanouil Benetos, and Thomas Pellegrini. Audio-jepa: Joint-embedding predictive architecture for audio representation learning. *arXiv preprint arXiv:2507.02915*, 2025.

Feng Wang, Sucheng Ren, Tiezheng Zhang, Predrag Neskovic, Anand Bhattad, Cihang Xie, and Alan Yuille. Vit-5: Vision transformers for the mid-2020s. *arXiv preprint arXiv:2602.08071*, 2026a.

Ziyu Wang, Kun Fang, and Yann LeCun. Music-jepa: Learning a world model of sound from action. *arXiv preprint arXiv:2607.22000*, 2026b.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

Haiyu Wu, Randall Balestriero, and Morgan Levine. Visreg: Variance-invariance-sketching regularization for jepa training. *arXiv preprint arXiv:2606.02572*, 2026.

Ho-Hsiang Wu, Prem Seetharaman, Kundan Kumar, and Juan Pablo Bello. Wav2clip: Learning robust audio representations from clip. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4563–4567. IEEE, 2022.

Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *Advances in neural information processing systems*, 34: 30392–30400, 2021.

Sihan Xu, Ziqiao Ma, Wenhao Chai, Xuweiyi Chen, Weiyang Jin, Joyce Chai, Saining Xie, and Stella X Yu. Next-embedding prediction makes strong vision learners. *arXiv preprint arXiv:2512.16922*, 2025.

Xiaoyu Yang, Yifan Yang, Zengrui Jin, Ziyun Cui, Wen Wu, Baoxiang Li, Chao Zhang, and Phil Woodland. Spear: A unified ssl framework for learning speech and audio representations. In *International conference on machine learning*, 2026.

Yumeng Yao, Jingzhi Dong, Haowen Gu, Tao Chen, Zonghan Wu, Xiaoshui Huang, and Yazhou Yao. Rethinking point clouds as sequences: A causal next-token predictive learning framework. *arXiv preprint arXiv:2605.17566*, 2026.

Goksenin Yuksel, Pierre Guetschel, Michael Tangermann, Marcel van Gerven, and Kiki van der Heijden. Wavjepa: Semantic learning unlocks robust audio foundation models for raw waveforms. *arXiv preprint arXiv:2509.23238*, 2025.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

A Related Work

Supervised Audio Pre-training. Supervised pre-training for audio has been mainly explored in two regimes. *Out-of-domain* approaches adapt models originally trained on labeled image datasets such as ImageNet (Deng et al., 2009) to spectrogram inputs, typically by modifying the input layer from three RGB channels to a single-channel spectrogram. Early work of this kind used CNN backbones such as EfficientNet (Gong et al., 2021b); more recent work adopts Transformer-based architectures, notably AST (Gong et al., 2021a) and HTS-AT (Chen et al., 2022a), which have driven substantial gains on audio classification benchmarks. *In-domain* approaches instead pre-train directly on the target modality. CLAP (Elizalde et al., 2023) adapts the CLIP (Radford et al., 2021) recipe to audio via contrastive language-audio alignment on supervised text-audio pairs, while Audio-CLIP (Guzhov et al., 2022) and Wav2clip (Wu et al., 2022) extend the CLIP backbone with an additional audio encoder trained on AudioSet (Gemmeke et al., 2017). Despite delivering strong results, these methods rely on large quantities of labeled data, which are expensive and time-consuming to obtain in practice.

Self-Supervised Audio Pre-training. Self-supervised learning has driven substantial progress in audio representation learning, largely by transferring ideas developed in the vision domain. Most recent methods extract log-mel spectrograms as input and follow the masked image modeling paradigm (He et al., 2022): Audio-MAE (Huang et al., 2022), MaskSpec (Chong et al., 2023) and MSM-MAE (Niizumi et al., 2022) directly apply the MAE reconstruction objective to spectrogram patches; and BEATs (Chen et al., 2022c) instead trains an iterative acoustic tokenizer to provide discrete semantic prediction targets. Waveform-based approaches such as wav2vec 2.0 (Baevski et al., 2020) and data2vec (Baevski et al., 2022) bypass spectrogram preprocessing altogether, and predict latent representations of the raw audio signal. More recent efforts push these directions further by combining ideas from multiple objectives. EAT (Chen et al., 2024) combines an MAE-style reconstruction with the data2vec latent-target formulation. A-JEPA (Fei et al., 2023) adapts the joint-embedding predictive paradigm to audio, using a target encoder updated via EMA and predicting masked patch embeddings. SSLAM (Alex et al., 2025) extends EAT with an additional source retention loss trained on artificially mixed audio to improve robustness to polyphonic content. Recently, SPEAR (Yang et al., 2026) unifies speech and general audio representation learning through the distillation of complementary knowledge from specialized teacher models. A few works have also focused on devising more sophisticated probing mechanisms to bridge the gap between fine-tuning and probing (Rauch et al., 2026; Ghaffari et al., 2026). Despite their diversity, these methods all rely on bidirectional prediction of masked or corrupted audio, and typically require a reconstruction decoder, a student-teacher setup, or auxiliary regularization losses to achieve competitive performance.

Next-token/embedding Prediction. Predictive learning has long been a central principle in representation learning across modalities. In language, GPT-style models (Radford et al., 2018; 2021; Achiam et al., 2023) established autoregressive next-token prediction as a scalable pre-training objective, with subsequent work confirming that the same paradigm transfers to vision (Image-GPT (Chen et al., 2020)) and beyond. More recent efforts move away from predicting raw signal tokens toward predicting embeddings directly. A first line of work adopts an *autoregressive* formulation: NEPA (Xu et al., 2025) introduces next-embedding predictive autoregression for visual representation learning; PointNTP (Yao et al., 2026) adapts causal next-token predictive learning to 3D point clouds; and LeWorldModel (Maes et al., 2026b) together with stable-worldmodel (Maes et al., 2026a) extend the paradigm to next-frame prediction for control tasks. A second line of work adopts a *joint-embedding* formulation: I-JEPA (Assran et al., 2023) and A-JEPA (Fei et al., 2023) predict latent representations of masked regions from context using dual encoders and an EMA teacher, while LeJEPA (Balestriero & LeCun, 2025) dispenses with masking and instead enforces invariance across augmented views with a variance-covariance regularizer. Together, these works suggest that predicting future tokens or embeddings can serve as a unified and scalable pre-training principle across modalities. In audio, however, this paradigm has remained largely unexplored: existing SSL methods rely almost exclusively on bidirectional masked modeling, and it is unclear *a priori* whether causal, per-position prediction is compatible with the non-stationary, temporally structured nature of audio signals. NAPE is, to the best of our knowledge, the first method to demonstrate that causal next-embedding prediction is a competitive and scalable pre-training objective for audio spectrograms, matching or surpassing more elaborate masked-modeling and joint-embedding methods with a substantially simpler recipe.

Table 6: **Hyperparameter list**. When an hyperparameter h varies between the base (h_b) and large (h_l) model, we include both values like (h_b/h_l). *Following (Chen et al., 2022c), we balance each class to 50% of the size of the unknown class for each training epoch.

Hyperparameters	**Pre-training**	**Fine-tuning**					
	AS-2M	*AS-2M*	*AS-20K*	*ESC-50*	*KS1*	*KS2*	*ER*
Optimizer	AdamW						
Opt. Momentum (β_1, β_2)	(0.9,0.95)	(0.9,0.999)					
Weight Decay	0.05						
Learning Rate Scheduler	Cosine Decay						
Layer-Wise LR Decay	1.0	0.7/0.9	0.8/0.9	0.7/0.9	0.7/0.8	0.7/0.8	0.7/0.9
Base Learning Rate	5e-3	1.25e-3					
Epochs	30/25	20/15	30/20	100	50	50	50
Warm-up Epochs	3	4/3	6/5	10	5	5	5
Batch Size	256/128	64					
GPUs	8	4	4	1	1	1	4
Weighted sampling	✗	✓	✗	✗	✓*	✗	✗
Multilabel	N/A	✓	✓	✗	✗	✗	✗
EMA Decay Rate	0.9999	0.99995	0.999	✗	✗	✗	✗
Label Smoothing	N/A	0.	0.	0.1	0.1	0.	0.1
Roll Augmentation	✗	✓	✓	✓	✗	✗	✓
Drop Path	0.	0.1					
SpecAug (time/freq)	N/A	(96,16)	(24,16)/(96,16)	(96,24)/(24,16)	(24,16)	(24,16)	(48,24)
Mixup (alpha/prob.)	N/A	(0.8,1.0)	(0.8,0.8)	(0.8,0.5)	(0.8,0.8)	(0.8,0.8)	(0.8,0.5)
Cutmix (alpha/prob.)	N/A	(1.0,1.0)	(1.0,0.8)	(1.0,0.5)	(1.0,0.8)	(1.0,0.8)	(1.0,0.5)
Noise Augmentation	✗	✓	✓	✓	✓	✓	✓
Loss Function	Neg Cos Sim	BCE	BCE	CE	CE	CE	CE
Dataset Mean for Norm.	-6.84	-6.84	-6.84	-6.84	-9.11	-9.16	-13.74
Dataset Std for Norm.	5.38	5.38	5.38	5.38	4.53	4.61	3.88

B Full Hyperparameter List

Fine-tuning Setting. We report the hyperparameter list for fine-tuning NAPE Base and Large in Table 6. Those values refer to the raster variant, with the diagonal variant having almost the same hyperparameters (the only difference is in the number of epochs needed to converge for *AS-20K*, which can vary of only a few epochs). Regarding NAPE Small, we used the same hyperparameters as NAPE Base, with the only difference being the value of the layer-wise learning rate decay ($LLRD$) for some of the downstream tasks as this hyperparameter depends on the number of layers of the model. Specifically, we set $LLRD = 0.6$ for *AS-2M* and $LLRD = 0.6$ for *KS1*/*KS2*.

Linear Probing Setting. For the linear probing experiments, we made the following changes (all other settings are the same as the fine-tuning hyperparameters): we increase the learning rate to $1e-2$ and we disable all augmentation techniques.

C Additional Ablation Studies

In this section, we report additional ablations studies on NAPE. For all experiments we use the raster scan variant. We use the SimSiam predictor for all ablations except for the ablation on the pre-training budget where we use the 2-MLP predictor.

Similarity Function ($\mathcal{D}$). In Table 7 we compare four choices for the similarity function $\mathcal{D}$ in Eq. 2: the *negative cosine similarity* (NAPE's default), a soft *cross-entropy* formulation that treats prediction and target as distributions over the d channels after applying a softmax (as in Chen & He (2021)), and the ℓ_1 and ℓ_2 distances between the two d-dimensional vectors.

Table 7: Ablation on the similarity function.

Similarity Function	**Audio Tasks**			**Speech Tasks**		
	AS-2M	*AS-20K*	*ESC-50*	*KS1*	*KS2*	*ER*
ℓ_1	**Diverge**					
ℓ_2	**Diverge**					
Cross-entropy	48.8	37.4	93.6	**98.0**	98.7	64.5
Cosine	**49.6**	**39.1**	**94.2**	97.9	**98.8**	**64.9**

Table 9: Ablation on freezing the emb. layer.

Freeze emb	*AS-2M*	*AS-20K*	*KS2*
✓	49.61	39.14	98.73
✘	49.58	39.08	98.80

Table 10: Ablation on positional encoding.

Positional Enc.	*AS-2M*	*AS-20K*
Absolute	47.8	36.7
RoPE	**49.6**	**39.1**

Both ℓ_1 and ℓ_2 cause pre-training to diverge: unlike cosine similarity, which is magnitude-invariant, these distance-based objectives can be trivially minimized by shrinking the norm of both predicted and target embeddings toward zero, a form of representation collapse in which the encoder outputs converge to a shared low-magnitude constant. The cross-entropy variant trains stably (softmax normalization implicitly bounds the target magnitude) but underperforms cosine on all benchmarks except the keyword-spotting tasks, where the two are on par. Cosine similarity, combining magnitude-invariance with a directionally informative loss signal, yields the best overall results and is retained as NAPE's default.

Patch Embedding Layer f. Table 8 compares three patchifiers, all configured to produce the same sequence length (504 tokens per 10s clip). The default *Conv2d* is a single strided convolution with kernel and stride 16×16, which treats time and frequency as symmetric 2D axes and matches the standard ViT design (Dosovitskiy et al., 2020). We also consider two alternatives motivated by observations from computer vision and speech literature. The *Convstem*, following Xiao et al. (2021), replaces the single Conv2d with four 3×3 stride-2 convolutions interleaved with batch normalization with the channel count growing progressively toward d; deeper convolutional stems have been shown to stabilize optimization and improve downstream performance in ViT-based image models. The *Speechstem*, inspired by the audio front-end of speech-oriented models such as Gemma 3n and Gemma 4 (Team et al., 2026), flattens the mel axis into feature channels and applies temporal-only 3×3 convolutions with a temporal downsampling factor of 2, producing a 1D time-only sequence of 504 tokens that reflects the frame-based processing typical of speech recognition systems.

Table 8: Ablation on the patch embedding layer.

Patch Emb. Layer	**Audio Tasks**			**Speech Tasks**		
	AS-2M	*AS-20K*	*ESC-50*	*KS1*	*KS2*	*ER*
Convstem	46.7	34.3	89.1	97.4	98.3	63.6
Speechstem	47.6	33.1	88.4	**98.1**	**98.9**	63.0
Conv2d	**49.6**	**39.1**	**94.2**	97.9	98.8	**64.9**

Conv2d outperforms Convstem across all benchmarks and Speechstem on 4 out of 6 tasks, with the largest gaps on AudioSet: on AS-20K, Convstem loses 4.8 mAP points and Speechstem loses 6.0 mAP spoints relative to Conv2d. Convstem's added non-linearity and speechstem's temporal-first inductive bias therefore do not translate into gains for our causal spectrogram prediction objective, treating frequency and time as symmetric 2D axes at the patchification stage is important for the pre-training signal that NAPE exploits. Conv2d is thus used as NAPE's default.

Freezing the Patch Embedding Layer. Table 9 examines whether freezing the patch embedding layer during fine-tuning affects downstream performance. In Xu et al. (2025), freezing the embedding layer yields significant improvements. However, in our setting, the two configurations are essentially indistinguishable on all three tasks tested. We therefore leave the patch embedding layer trainable during fine-tuning, matching the standard practice of prior audio Transformer methods.

Positional Encoding. Table 10 compares learned absolute positional encodings with Rotary Position Embedding (RoPE) (Su et al., 2024), applied independently along the frequency and time axes. RoPE substantially outperforms absolute encodings on both AudioSet benchmarks, with gains of $+1.8$ mAP on AS-2M and $+2.4$ mAP on AS-20K. Beyond the accuracy improvement, RoPE also offers a practical benefit: because it encodes relative positions rather than absolute ones, it transfers naturally to clips of different lengths at inference time, whereas absolute encodings require interpolation. We therefore adopt RoPE as NAPE's default positional encoding.

Attention Type and Pooling. Table 11 jointly ablates the attention mask used during fine-tuning and the pooling strategy for producing the clip-level representation. Under causal attention, the natural pooling choice is the last token: only the final position has access to the full preceding context, and the CLS token (added during fine-tuning) functions more as a BOS-like anchor at the start of the sequence than as a readout summary of the clip (this is in line with (Xu et al., 2025)). Under bidirectional attention, however, every position sees the entire sequence, so CLS, last-token, and mean pooling all in principle have access to the same information. This is reflected in the numbers: the three bidirectional variants are essentially on par on the three tested tasks. Keeping causal attention with last-token pooling remains competitive but slightly worse than its bidirectional counterpart, with a 0.2 mAP drop on both AS-2M and AS-20K. We therefore use bidirectional attention with mean pooling as NAPE's default fine-tuning configuration.

Table 11: Ablation on the pooling method and attention type.

Attention Type	Pooling Mode	Task		
		AS-2M	*AS-20K*	*KS2*
Bidirec	CLS Tok	**49.7**	38.9	98.7
Bidirec	Last Tok	49.6	38.7	**98.8**
Bidirec	**Avg Pool**	49.6	**39.1**	**98.8**
Causal	Last Tok	49.4	38.9	**98.8**

Normalization Layer. Table 12 compares LayerNorm (Ba et al., 2016) (LN) and RMSNorm (Zhang & Sennrich, 2019) in the encoder. RMSNorm drops the mean-centering step of LayerNorm and normalizes each feature vector by its root-mean-square, which has become common in recent language and vision Transformers. In our setting, the two variants are effectively on par, with small variations across the six downstream tasks. For this reason, we retain LayerNorm as NAPE's default for consistency with prior audio Transformer works.

Table 12: Ablation on normalization layer.

Norm Type	Audio Tasks			Speech Tasks		
	AS-2M	*AS-20K*	*ESC-50*	*KS1*	*KS2*	*ER*
LN	49.6	39.1	94.2	97.9	98.8	64.9
RMS	49.5	38.8	94.4	97.8	98.8	65.2

Pre-training Budget. Figure 8 reports downstream mAP on AS-2M and AS-20K when fine-tuning from NAPE-Base and NAPE-Large checkpoints obtained after 6, 12, 18, and 30 epochs of pre-training. Performance improves monotonically with the pre-training budget for both scales and both benchmarks: on AS-2M, NAPE-Base improves from 48.49 mAP after 6 epochs to 49.39 after 30; on AS-20K, from 36.51 to 38.50. Notably, NAPE already delivers strong results after only a few epochs of pre-training. After just 6 epochs, NAPE-Large reaches 49.45 mAP on AS-2M, already matching or exceeding most published baselines (cf. Table 4), making NAPE an appealing choice under tight compute budgets. Longer pre-training continues to yield returns at 30 epoch, and NAPE-Large consistently benefits more than NAPE-Base from a larger budget, consistent with the scaling behavior discussed in Section 3.3.

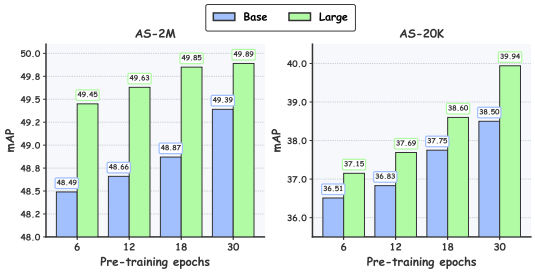


Figure 8: Pre-training budget ablation. AS-2M (left) and AS-20K (right) mAP trend as a function of the number of epochs the model is trained on.

Random Masking. Table 13 examines whether adding random masking to the input embeddings during pre-training, in the style of masked modeling methods, helps NAPE. We consider three masking ratios: no masking (0%, the default), 20%, and 50%. We use NAPE small with raster scan order. Across all six benchmarks, masking either has no effect or slightly degrades downstream performance, with the largest drop on AS-20K (-1.0 mAP at 50%) and ESC-50 (-1.2 points at 50%). This differs from methods such as Audio-MAE (Huang et al., 2022) or BEATs (Chen et al., 2022c), where high masking ratios are central to the pre-training signal: the model is asked to reconstruct or predict the masked content from the visible context. In NAPE, the causal attention

Table 13: Ablation on random masking applied to the input embeddings.

Mask Ratio	Audio Tasks			Speech Tasks		
	AS-2M	*AS-20K*	*ESC-50*	*KS1*	*KS2*	*ER*
0	**47.6**	**36.2**	**92.9**	97.4	98.3	**64.6**
20	47.3	**36.2**	92.2	**97.6**	**98.5**	63.9
50	47.2	35.2	91.7	97.5	98.3	63.9

mask already prevents each position from accessing its target, so additional input masking removes useful context without changing the difficulty of the prediction task. We therefore leave the input unmasked in NAPE's default configuration.

D NAPE'S PSEUDOCODE

Algorithm 1 lists a NAPE pre-training step in PyTorch-like pseudocode. Each batch of log-mel spectrograms `x` is passed through the patch embedding layer `f`, producing the sequence of patch embeddings `z`. The scanning order is applied inside `f`, when the 2D patch grid is flattened into a sequence of length `N`. The same `z` is then used twice, as the input to the causal Transformer `h` and as the source of the prediction targets, so one forward pass of `f` per step suffices and no separate target network is required. The predictor `g` maps the encoder outputs to the predicted embeddings `z_hat`.

Algorithm 1 NAPE pre-training algorithm.

```
# f: Patch Embedding Layer
# h: Causal Transformer Encoder
# g: Predictor

for x in loader:          # x: [B,1,F,T_frames]
    z = f(x)              # embeddings [B,N,D]
    z_hat = g(h(z))       # predictions [B,N,D]
    loss = D(z, z_hat)
    loss.backward()
    update(f,h,g)

def D(z, z_hat):
    target = z[:,1:,:].detach() # stop-grad
    pred = z_hat[:,:-1,:]       # AR shift
    pred = normalize(pred, dim=-1)
    target = normalize(target, dim=-1)
    return -(pred*target).sum(-1).mean()
```

The distance function `D` implements the autoregressive shift and the stop-gradient. Targets are read from positions `1:` and detached, predictions from positions `:-1`, so the prediction made at position t is compared against the embedding at position $t+1$ and no gradient reaches the target branch. Both are ℓ_2-normalized and the loss is their negative dot product averaged over the `N-1` valid positions, recovering the negative cosine similarity of Eq. 2. `f`, `h` and `g` are trained jointly; `g` is used only during pre-training, and downstream adaptation retains `f` and `h` alone plus a task-dependent classification head.

E PRE-TRAINING COST COMPARISON ACROSS METHODS

Table 14 compares NAPE's pre-training cost against four audio SSL methods in terms of wall-clock following EAT (see Table 2 in (Chen et al., 2024)). NAPE-B is the least expensive full-budget entry at 208 GPU-hours, an order of magnitude below Audio-MAE (2304) and BEATs$_{iter3}$ (3600), while obtaining a stronger AS-2M result than either. The 3090-based methods are the closer comparisons in hardware generation. EAT sits nearest to NAPE-B in cost (230 GPU-hours), ahead on AS-20K (40.2 vs 39.1) and behind on AS-2M (48.6 vs 49.6). SSLAM, at 430 GPU-hours, is closely matched in cost to NAPE-L (416): the two tie on AS-2M at 50.2 mAP, with SSLAM slightly ahead on AS-20K (40.9 vs 40.5) using a third of the parameters. Most striking is the short-budget entry: a NAPE-L checkpoint pre-trained for only 6 epochs costs 100 GPU-hours, less than half of any other row, and already reaches 49.7 mAP on AS-2M, exceeding every baseline except SSLAM. AS-20K benefits considerably more from a longer budget (Figure 8) and trails at 37.7.

F NAPE VS JEPA VS LEWORLDMODEL FORMULATION

F.1 NAPE VS JEPA-STYLE METHODS

NAPE and JEPA methods Assran et al. (2023); Fei et al. (2023) learn representations by predicting one set of patch embeddings from another, therefore it is worth clarifying how the two paradigms differ structurally. Figure 9 (left and right) illustrates the two frameworks.

JEPA-style methods use *two* encoders: a context encoder h that processes the visible portion of the input x, and a target encoder that produces the prediction targets from the complementary region y. The target encoder is not trained by gradient descent; instead, its weights are updated as an EMA of the context encoder's weights, so that its output remains a moving target that the predictor tries to match. The predictor itself is a Transformer that receives both the context representation and an additional variable c, which encodes the target positions to be predicted, and produces the predicted target embeddings $\hat{z}_y$ in parallel. The context encoder operates with bidirectional attention over

Table 14: **Comparison of pre-training cost across multiple SSL methods**. Costs for the other methods are as reported in the respective papers; NAPE's wall-clock time is measured on our own hardware (26 hours for NAPE-B, 52 for NAPE-L). GPU-hours are not normalized across hardware generations or implementations. Speedup is relative to BEATs_{iter3}.

Model	Hardware	#Par.	Epochs	GPU-h	Speedup	AS-2M	AS-20K
$\text{BEATs}_{\text{iter3}}$	16×V100	90M	342	3600	1.0×	48.0	38.3
Audio-MAE	64×V100	86M	32	2304	1.6×	47.3	37.1
SSLAM	4×RTX 3090	88M	15	430	8.4×	50.2	40.9
EAT	4×RTX 3090	88M	10	230	15.7×	48.6	40.2
NAPE-B	8×L40S	85M	30	208	17.3×	49.6	39.1
NAPE-L	8×L40S	303M	6	100	36.0×	49.7	37.7
NAPE-L	8×L40S	303M	25	416	8.7×	50.2	40.5

the visible patches. To prevent representation collapse under this regime, JEPA methods typically require additional machinery beyond stop-gradient: an EMA teacher (I-JEPA (Assran et al., 2023), A-JEPA (Fei et al., 2023)), auxiliary regularizers such as LeJEPA (Balestriero & LeCun, 2025) and VISReg (Wu et al., 2026), or masking strategies carefully tuned so that the two views do not degenerate into trivial solutions.

NAPE, in contrast, uses a *single* encoder (h) and takes its prediction targets directly from the shared patch embedding layer f. Because f is a shallow, non-recurrent module, its outputs provide a stable target signal that does not need to be maintained by a separate EMA-tracked encoder. NAPE also features a lightweight MLP head (g) that operates on a single position at a time, rather than a Transformer that jointly reasons about context and target positions: a structurally simpler design that reflects NAPE's per-position causal prediction task. There is no masking: the model processes the full input as a causal sequence, and predictions are made one patch at a time under a causal attention mask. Collapse prevention rests on the three key components discussed in Section 2.2, causality, prediction shift, and stop-gradient, together with the magnitude-invariance of the cosine objective (Section 3.2), and requires no auxiliary regularizers or student-teacher setups.

F.2 NAPE vs LeWorldModel Methods

A closely related recent method is LeWorldModel (Maes et al., 2026b), a JEPA designed for world modeling that also autoregressively predicts next-frame embeddings from raw pixels (Figure 9, middle). Like NAPE, LeWM aims to simplify the JEPA recipe and dispenses with EMA teachers and pre-trained encoders, training its encoder and predictor jointly end-to-end. Unlike NAPE, LeWM relies on the SIGReg regularizer (Balestriero & LeCun, 2025), an auxiliary loss that projects the learned embeddings onto random directions and enforces Gaussian-distributed marginals, to prevent collapse, whereas NAPE requires only stop-gradient on the target branch. LeWM uses MSE as its prediction loss (kept stable by SIGReg), whereas NAPE uses cosine similarity combined with stop-gradient, which together avoid the shrink-to-zero collapse mode observed with ℓ_1 and ℓ_2 under a magnitude-sensitive loss (Section 3.2). Finally, the two methods target different problems: LeWM predicts across time steps of an action-conditioned trajectory for planning in latent space, while NAPE predicts across patches within a single input for representation learning applicable to classification tasks. NAPE therefore constitutes a distinct point in the JEPA design space: one that shares LeWM's end-to-end simplicity but arrives at collapse prevention through a fundamentally different mechanism.

G Training Loss Visualizations

To better understand NAPE's training dynamics, in Figure 10 we report the pre-training loss curves of six NAPE runs across multiple scales and configurations. The top row reports runs of the default NAPE configuration (raster scanning, SimSiam predictor, cosine similarity, all three key components enabled) at the three model scales: **Small** (top left), **Base** (top middle), and **Large** (top right). All three curves follow the same qualitative trajectory: a rapid descent from around -0.3 to below -0.9 within the first $\sim 20{,}000$ steps, followed by a smooth and steady decrease that asymptotes just below

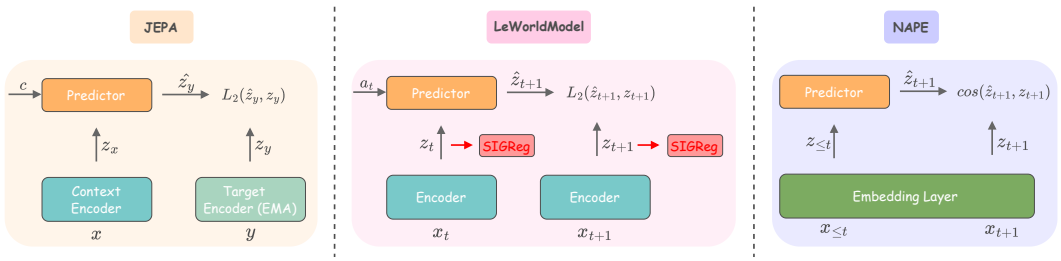


Figure 9: Comparison between JEPA, LeWorldModel, and NAPE architectures.

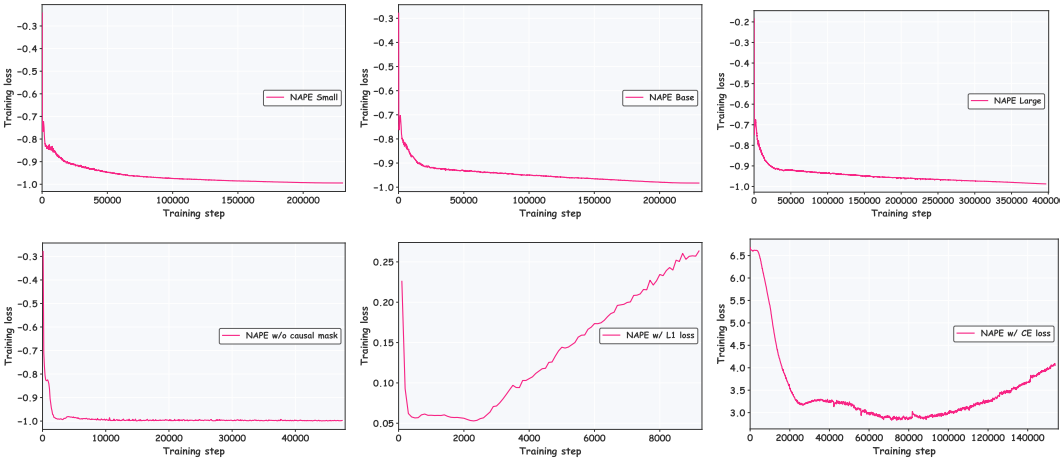


Figure 10: Pre-training loss curves for multiple NAPE models across scales and configurations.

-0.98, close to the theoretical minimum of -1. The Large model runs for roughly twice as many steps as Small and Base, its batch size is halved to fit into memory, but reaches a comparable final loss along a trajectory of the same shape, indicating that the NAPE objective is well-conditioned across encoder capacities.

The bottom row reports three ablation runs at the Base scale. **Without the causal mask** (bottom left), the loss collapses almost immediately, saturating near -1.0 within $\sim 2{,}000$ steps and remaining there. The model has found the trivial identity mapping enabled by bidirectional attention: the objective is satisfied without learning any structure that transfers downstream, as reflected in the sharp mAP drop reported in Table 1.

The configuration with **L1 loss** (bottom middle) diverges: the loss drops rapidly to ~ 0.05 within a few hundred steps, plateaus briefly, and then climbs steadily. The initial descent reflects the model finding low-norm solutions in which both predicted and target embeddings shrink toward zero, since L1 is magnitude-sensitive and trivially reduced by scaling the outputs; once the encoder approaches this degenerate regime, the objective becomes ill-conditioned and the loss reverses. The **cross-entropy loss** (bottom right), in contrast, converges to a reasonable minimum around step $\sim 70{,}000$ before gradually overfitting. Unlike L1, CE does not collapse: this suggests that collapse prevention is not exclusive to cosine similarity. In our setting, however, CE still yields worse downstream performance than cosine (Table 7), and its tendency to overfit after the initial descent makes it a less robust choice for NAPE pre-training.

H Additional Qualitative Results

We include additional qualitative results for NAPE-L with raster order in Figure 11 and NAPE-L with diagonal order in Figure 12.

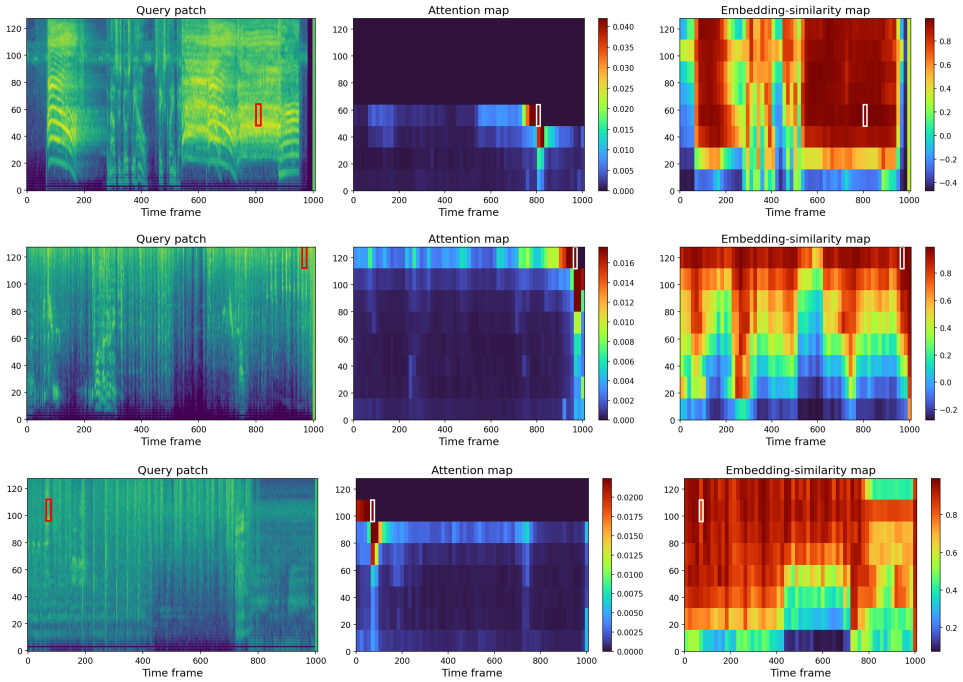


Figure 11: Additional qualitative results for NAPE-L with raster order.

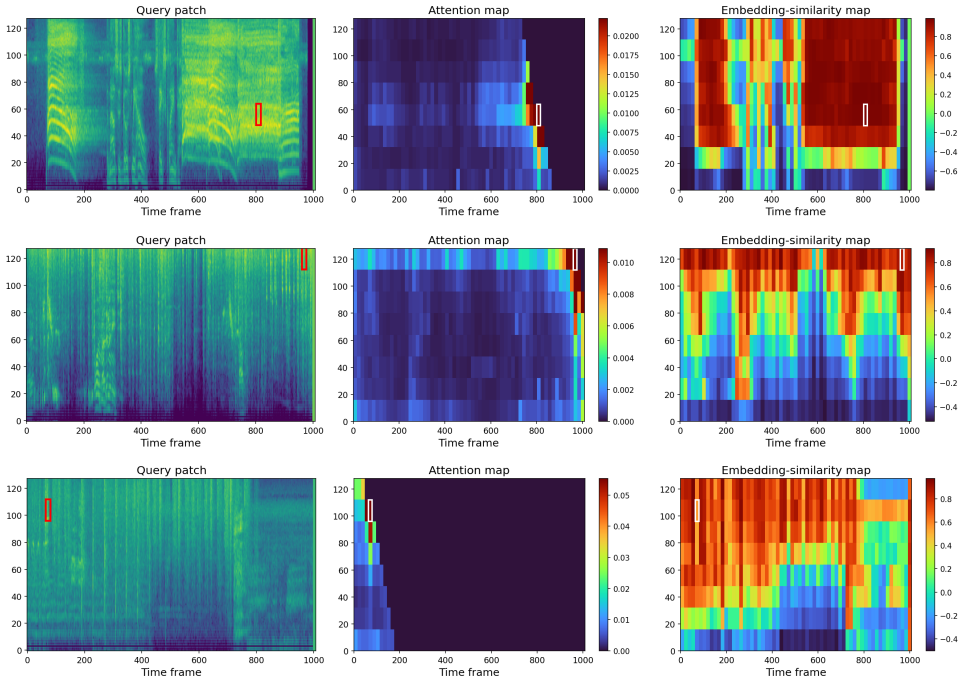


Figure 12: Additional qualitative results for NAPE-L with diagonal order.

I LIMITATIONS

NAPE processes the full patch sequence during pre-training, whereas masked autoencoders such as Audio-MAE (Huang et al., 2022) feed only the unmasked subset through the encoder, a substantial per-step saving at equal model size. Nor is this a cost NAPE can trade away by masking, as Table 13 shows that masking the input embeddings does not improve downstream performance. In total,

Table 14 shows NAPE to be the cheaper method by an order of magnitude, since it converges in far fewer epochs; the higher per-step cost would nonetheless become the binding constraint at longer sequence lengths, which is the regime a speech-oriented instantiation would require (Appendix J).

Next-embedding prediction optimizes for continuing a sequence rather than for producing linearly separable features, and the two objectives are only loosely aligned. Settings that rely on frozen features rather than fine-tuning may therefore be better served by objectives that target separability directly.

We evaluate NAPE on the six benchmarks established by prior work in audio SSL, which allows direct comparison against strong published baselines under a common protocol. Whether pre-training NAPE directly on speech yields representations suited to tasks such as speech recognition therefore remains untested, since both the pre-training corpus and the required temporal granularity differ substantially from our setting. Appendix J discusses what such an extension would involve.

J Extending NAPE to Speech and Multimodal Settings

Following common practice in audio SSL, we evaluate NAPE on the benchmarks established by prior work (Huang et al., 2022; Chen et al., 2022c; 2024; Alex et al., 2025). A natural direction beyond this setting is speech representation learning, with automatic speech recognition (ASR) and audio-visual speech recognition (AVSR) as representative cases. We outline below the aspects of NAPE's design such an extension would revisit.

Fine-tuning an AudioSet-pre-trained encoder for ASR would compare a general audio model against speech-specialized ones (Baevski et al., 2020; Hsu et al., 2021; Chen et al., 2022b), and any gap would be attributable to the pre-training corpus rather than to the objective. A meaningful evaluation requires pre-training NAPE on speech at comparable scale, which is beyond the compute budget of this work. Regarding the patch granularity, NAPE's 16×16 patches span 160 ms, coarser than the phonetic units ASR must resolve. The *speechstem* patchifier (Table 8) is a first step toward a speech-oriented tokenization, producing a purely temporal token sequence and matching the Conv2d default on the keyword-spotting tasks. It also collapses the scanning order to the temporal ordering, removing a design axis specific to NAPE.

Our ablations identify the prediction target as the component that most constrains what NAPE learns. Predicting the next patch embedding rewards short-term spectral continuity, which ASR must instead abstract over. For the AVSR extension (Ma et al., 2023; Shi et al., 2022; Cappellazzo et al., 2025; 2026a;b), audio patches and visual frames arrive at different rates and must be interleaved into a single causal sequence, the multimodal counterpart of the scanning order studied here. We leave these directions to future work.