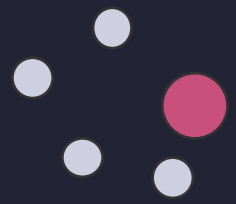


HuBERT

Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units

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

IEEE/ACM Transactions on Audio, Speech and Language Processing, 29, 3451–3460 (2021) · arXiv:2106.07447

Masked prediction and full-stack pretraining · Talk 1 of 2 · 45 minutes

Why this paper matters

The model that became the default speech backbone

100

k-means clusters on MFCCs — the entire first teacher

2

iterations of clustering to reach state of the art

964M

parameters in the X-LARGE model

13%

relative WER reduction on test-other from scaling to 1B

The claim

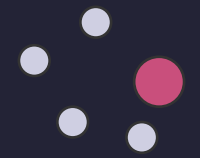
Run k-means on MFCC features to get crude frame labels. Train a BERT to predict those labels, but only on masked frames. Then re-cluster using the model's own intermediate representations and repeat. Two rounds of that matches or beats wav2vec 2.0 across every labelled-data setting.

The idea underneath it

HuBERT relies primarily on the consistency of the unsupervised clustering step rather than the intrinsic quality of the assigned cluster labels. The teacher can be almost worthless as a phonetic classifier and still be a perfectly good training signal — provided the model has to infer its labels from context rather than copy them.

1 Three problems

Why masked prediction does not transfer directly from text



The three problems, as the paper states them

Every design decision in the method answers one of these

Multiple sound units per utterance

Computer vision pre-training often treats each image and its augmentations as one class to be pulled together. An utterance contains dozens of different sounds, so instance classification is meaningless here.

No lexicon of sound units

In NLP the vocabulary is given — words or word pieces. In speech there is no inventory to predict over, so a predictive loss has nothing to predict.

No known segmentation

Sound units have variable lengths and the boundaries between them are unknown, which complicates masked prediction: you cannot mask 'a phoneme' because you do not know where it starts.

HuBERT's answer to all three, in one sentence

Manufacture a lexicon by clustering frames, which gives you a finite inventory and a label for every 20 ms frame — so segmentation stops mattering, because you predict frame labels rather than unit identities.

The remaining question is whether an inventory this crude can teach a model anything. That is what section 4 is about.

What HuBERT is reacting against

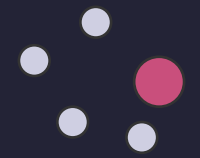
The paper's own criticism of wav2vec 2.0

Careful negative sampling	The contrastive loss requires deciding where to draw distractors from — a design choice with a large effect and no principled answer.
An auxiliary diversity loss	A second loss term with a tuned weight, needed purely to stop the learned codebook from collapsing.
A temperature schedule	The Gumbel softmax needs a properly annealed temperature. Get the schedule wrong and the discrete choice is either never made or made arbitrarily.
Quantising the wrong thing	wav2vec 2.0 only quantises the convolutional encoder output, which may not be the best feature to quantise given the limited capacity of that encoder.

HuBERT's proposal: separate acoustic unit discovery from representation learning, and use a plain cross-entropy loss.

2 The method

Offline clustering, masked prediction, iterative refinement



The core idea

Two stages, alternating

Why the loop terminates somewhere useful

A pre-trained model should provide better features than raw MFCCs. So clustering its representations produces a better teacher than the one it was trained on, and the next model trained on that teacher is better still.

The authors call this an extension of iterative pseudo-labelling to the self-supervised setting. Two rounds are enough to reach state of the art.

Architecture and loss

Deliberately identical to wav2vec 2.0 where possible

Architecture

The same convolutional waveform encoder: seven blocks, 512 channels, strides (5,2,2,2,2,2,2), kernel widths (10,3,3,3,3,2,2). One frame per 20 ms of 16 kHz audio.

Then a Transformer, a projection layer, and a code embedding layer. Nothing else.

The prediction head

The Transformer output at step t is projected, then scored against every codeword embedding by cosine similarity, divided by a temperature of 0.1, and passed through a softmax.

So the model learns an embedding for each cluster rather than a plain classifier weight — the same shape as the wav2vec 2.0 target space, reached by a different route.

The loss

$L = \alpha \cdot L_m + (1 - \alpha) \cdot L_u$, where L_m is the cross-entropy summed over masked timesteps and L_u over unmasked ones. Masking follows SpanBERT and wav2vec 2.0: p% of timesteps are sampled as span starts and l consecutive frames are masked. HuBERT uses p = 8% and l = 10.

α is the paper's most important hyperparameter and its most interesting experiment. α = 1 means masked frames only; α = 0 means unmasked frames only, which reduces to imitating the clustering model.

Why predict masked frames only

The conceptual argument, before the numbers

α = 0 · unmasked frames

The model sees the frame and predicts its own label. It can succeed by learning the clustering function itself.

That caps the model at the teacher's quality: it is being asked to imitate k-means, and it cannot do better than what k-means knows.

α = 1 · masked frames

The frame is hidden. The only way to predict its label is from surrounding context.

This forces two things at once: representing unmasked audio well, which is acoustic modelling, and capturing long-range temporal structure between those representations, which is language modelling.

Why this makes the label noise tolerable

If the teacher assigns the same wrong label consistently to the same sound, a model predicting from context can still learn the sequential structure of the language — it just learns it over a relabelled alphabet. Systematic error is survivable; random error is not.

That is what 'consistency rather than correctness' means, and it is why a teacher with a phone purity around a third can still bootstrap a state-of-the-art model.

Two ways to improve the teacher

Both are optional; only one is essential

Cluster ensembles

Use several clustering models at once and predict all their labels, one projection head each. This is multi-task learning with tasks invented by unsupervised clustering.

An ensemble of k-means models with different codebook sizes creates targets at different granularities — from manner classes such as vowel versus consonant, down to sub-phone states.

It also combines with product quantisation: split the feature space into subspaces and quantise each separately.

Iterative refinement

Cluster the learned representations of the previous iteration's model, then retrain on the new labels.

Iteration 1: k-means with 100 clusters on 39-dimensional MFCCs — 13 coefficients plus first and second derivatives.

Iteration 2: k-means with 500 clusters on the 6th Transformer layer output of the iteration-1 BASE model, fitted on a 10% sample of the data.

Which layer, and why it changes between iterations

Iteration 1's best features are in the middle of the network, around layer 6; its last layers degrade sharply, because they have learned to mimic a bad teacher.

Iteration 2's features improve monotonically with depth, so the LARGE and X-LARGE models are trained on labels from layer 9 of the iteration-2 BASE model — making them effectively third-iteration models.

Three model configurations

Same convolutional encoder throughout

	BASE	LARGE	X-LARGE
Transformer layers	12	24	48
Embedding dimension	768	1,024	1,280
Feed-forward dimension	3,072	4,096	5,120
Attention heads	8	16	16
Layer-drop probability	0.05	0	0
Projection dimension	256	768	1,024
Parameters	95M	317M	964M
Pre-training data	LibriSpeech 960h	Libri-Light 60k h	Libri-Light 60k h
GPUs	32	128	256
Audio per GPU per batch	≤ 87.5 s	56.25 s	22.5 s
Peak learning rate	5×10^{-4}	1.5×10^{-3}	3×10^{-3}
Update steps	250k then 400k	400k	400k

BASE is trained for two iterations; LARGE and X-LARGE for one iteration each, using labels from the second-iteration BASE model.

Fine-tuning

Where the labels finally enter

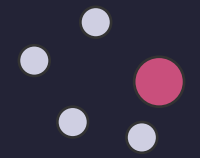
Output layer	The projection layers are removed and replaced with a randomly initialised softmax. The CTC vocabulary is 26 English characters, a space, an apostrophe and a blank symbol.
Frozen encoder	The convolutional waveform encoder is fixed throughout fine-tuning. Only the Transformer and the new output layer update.
Freeze step	As in wav2vec 2.0, a hyperparameter controls how many steps the Transformer stays frozen while only the softmax matrix trains.
Model selection	Peak learning rate, schedule, number of steps, freeze step and masking probability are all swept per model size and per labelled split, selected on dev-other word error rate.
Decoding	Beam search with an n-gram or Transformer language model, with weights tuned by Bayesian optimisation — the same setup as

Note the sweep in row four: the reported numbers are the best of a per-setting hyperparameter search. Fair, standard, and worth remembering when reading small margins.

3

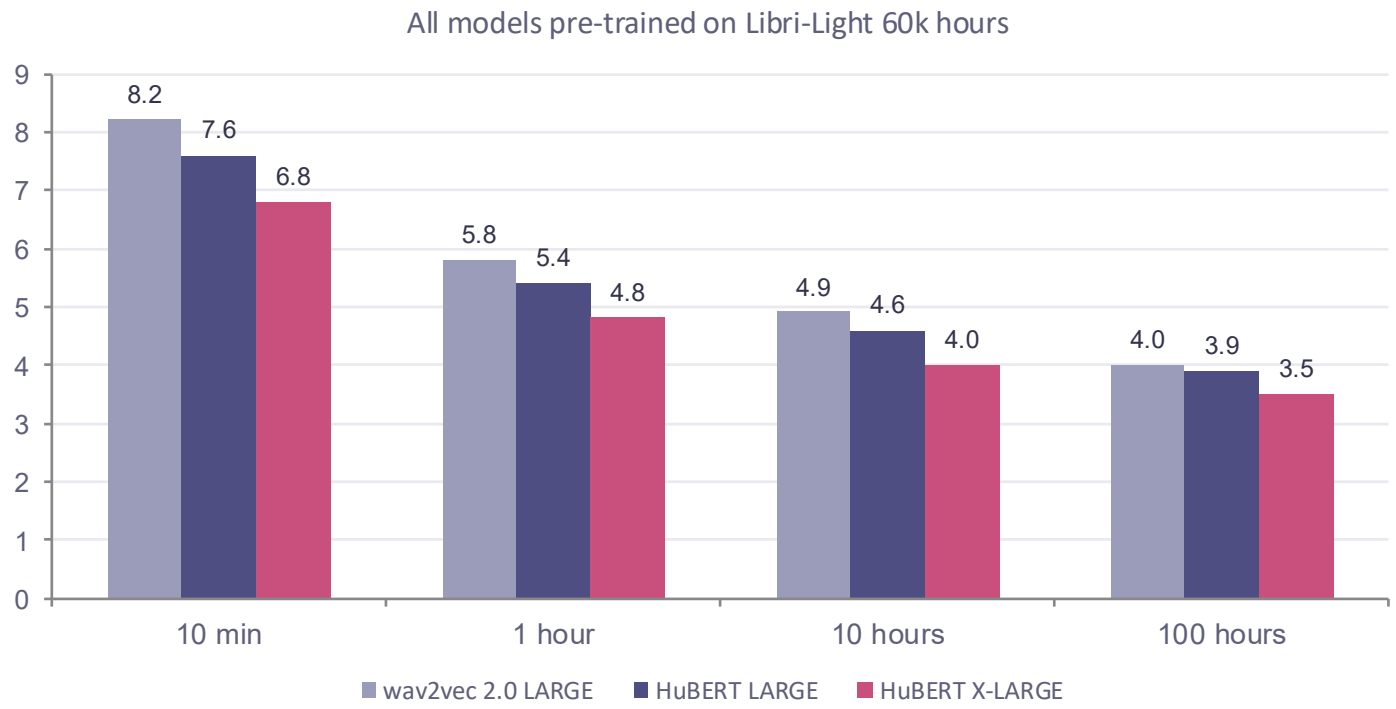
Does it work?

Low-resource and high-resource speech recognition



Low-resource results

Word error rate on test-other, decoded with the Transformer language model

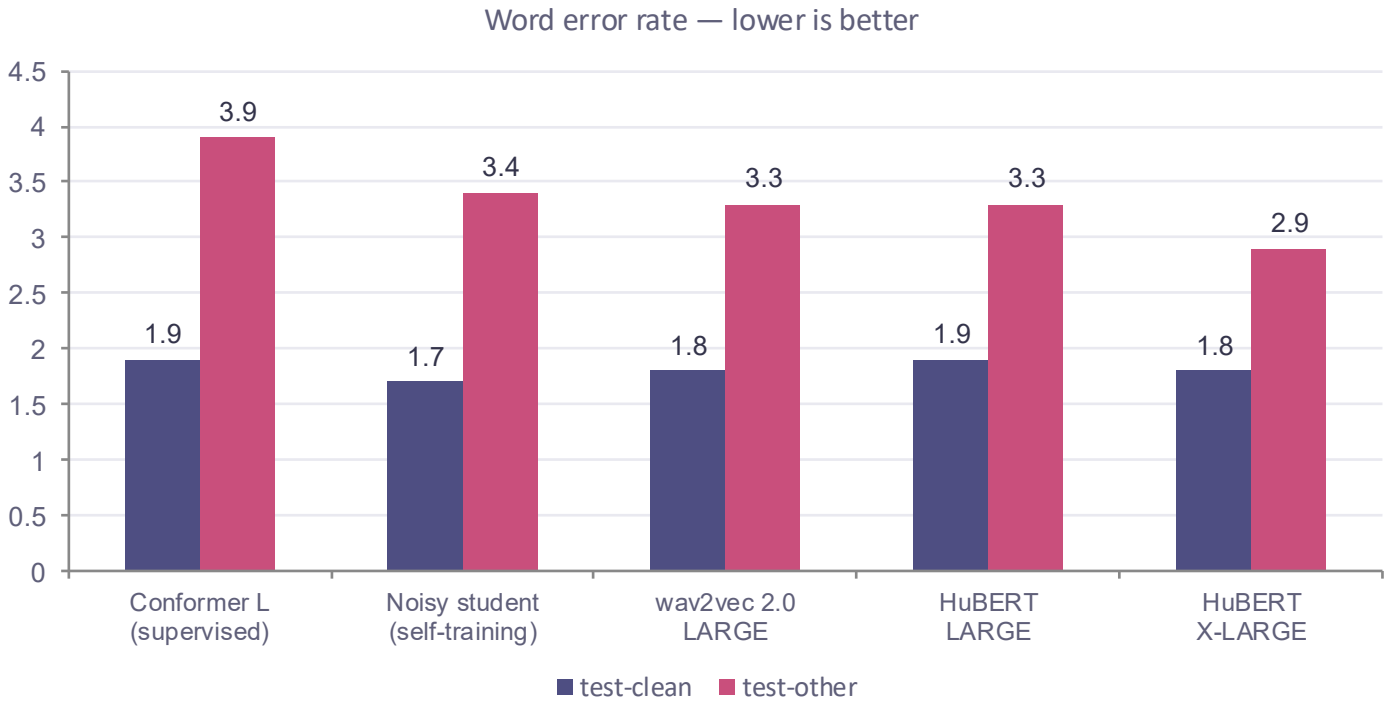


- With ten minutes of labels, HuBERT LARGE reaches 4.7 on test-clean and 7.6 on test-other — 0.1 and 0.6 better than wav2vec 2.0 LARGE.
- Scaling to a billion parameters takes test-other to 6.8, a further substantial gain.
- HuBERT beats Discrete BERT by a wide margin in every setting, despite a virtually identical objective — because it takes raw waveform input rather than quantised units.
- Two exceptions the paper reports rather than hides: at 100 hours, LARGE is 0.1 worse on test-clean, and BASE is 0.1 worse on test-other.

Read the margins honestly: HuBERT matches or slightly beats wav2vec 2.0. It does not transform the numbers. The interesting claim is that it does so with a simpler and more stable recipe.

High-resource results

Fine-tuning on all 960 labelled hours of LibriSpeech

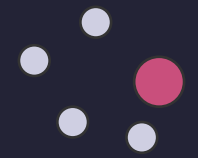


- HuBERT LARGE ties wav2vec 2.0 LARGE at 3.3 on test-other. X-LARGE takes it to 2.9.
- Both beat the best supervised and self-training systems, but lag methods that combine pre-training with self-training.
- The paper says plainly that it expects HuBERT would close that gap if combined with self-training, since its pre-trained model is at least as good as the ones those methods start from.
- The X-LARGE gain — 13% relative on test-other over LARGE — is the paper's evidence that the method scales.

A prediction that aged well: pre-training and self-training are complementary, and combining them was the obvious next step. It became standard practice within about a year.

4 Why it works

The analyses — the most valuable part of the paper



How do you measure a teacher you never supervised?

Three metrics, computed against forced-aligned phone labels used only for analysis

Phone purity	Label each cluster with its most likely phone, then ask how often frames in that cluster really are that phone. Effectively frame-level phone accuracy — but it goes to 100% if every frame gets its own cluster, so it only compares like with like.
Cluster purity	The mirror image: for each phone, how often do its frames land in that phone's most likely cluster? Falls as the number of clusters rises.
Phone-normalised mutual information	The information-theoretic version: what fraction of the uncertainty about the phone label is eliminated by knowing the cluster? Comparable across different numbers of units, which is why it is the workhorse metric here.

The alignments come from a supervised hybrid recogniser and are used only to analyse the teachers, never to train. This is a good pattern: build the system without labels, then use labels to understand what it learned.

How bad is the first teacher?

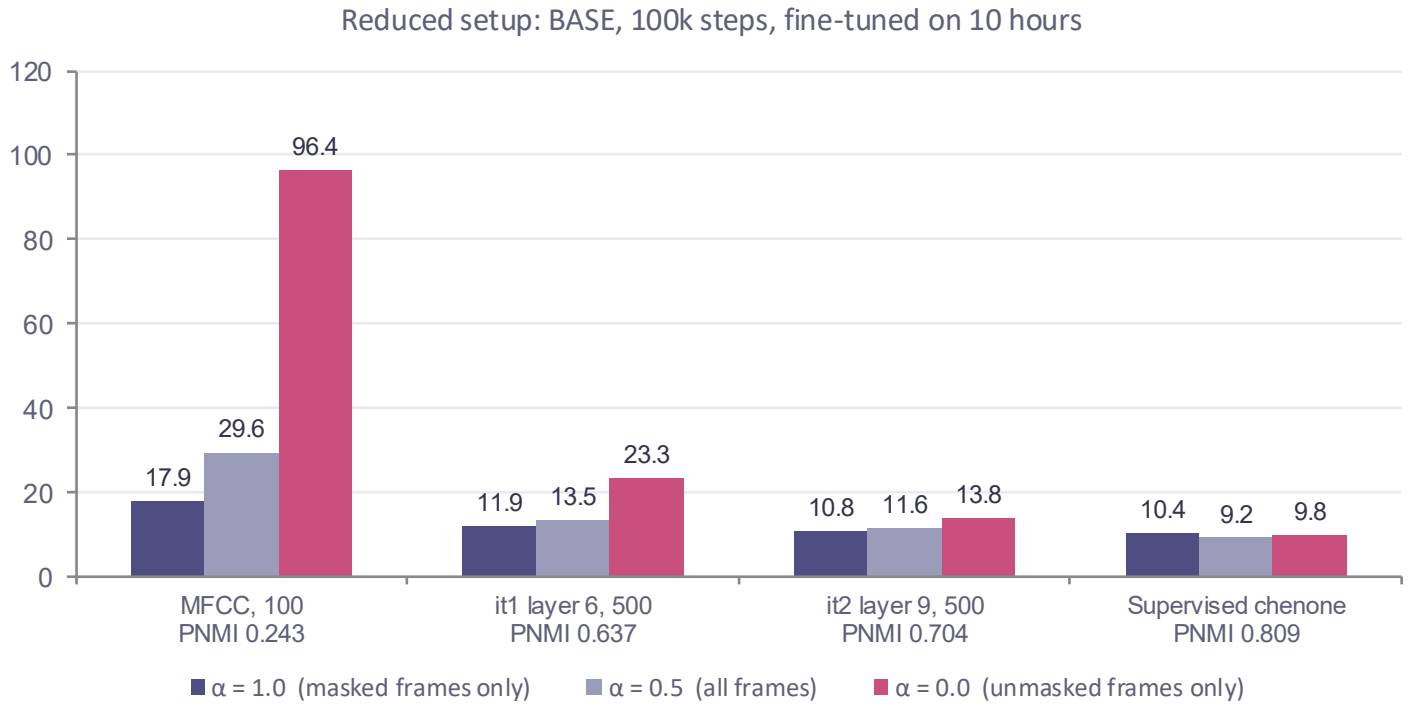
Phone-normalised mutual information — how much it knows about phones

- The starting teacher removes about a quarter of the uncertainty about phone identity. It is a genuinely poor phonetic classifier.
- One round of refinement more than doubles that. A second round adds a little more.
- Refined clusters approach the supervised topline, which uses nearly nine thousand context-dependent grapheme units.
- K-means is also stable: across ten trials the standard deviation is at most about 0.012, and fitting on 1 hour versus 100 hours changes PNMI by roughly 0.01.

The stability result matters practically: you can fit the clustering on a small subsample and lose almost nothing, which is what makes the pipeline affordable.

The experiment that makes the argument

Word error rate on dev-other for three settings of α, across four teachers

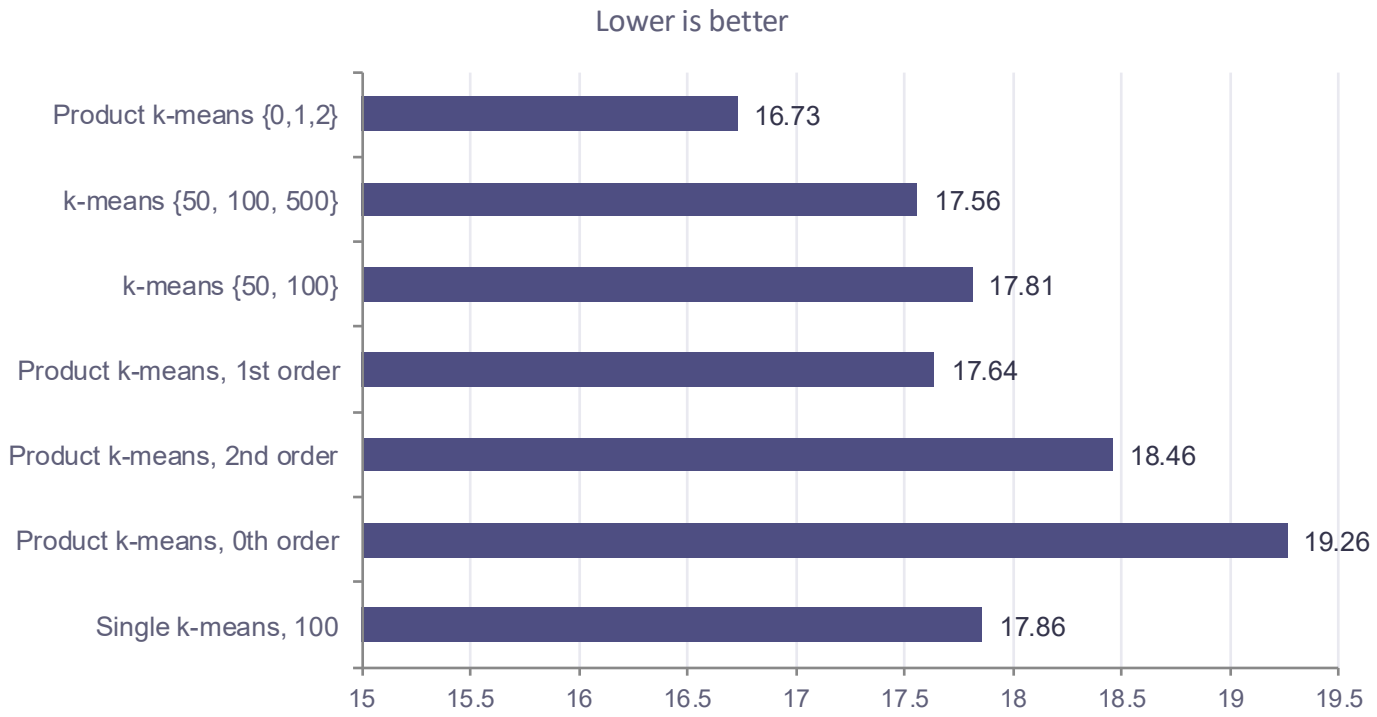


- With the worst teacher, predicting unmasked frames gives 96% word error — the model has learned to imitate k-means and nothing else.
- The same teacher with masked-only prediction gives 17.9%. Identical labels, identical architecture, one hyperparameter.
- As the teacher improves, the penalty for the unmasked loss shrinks — 23.3 falls to 13.8.
- With genuinely good supervised targets the ordering reverses: α = 0.5 beats α = 1.0.

This single table is the paper. Masked-only prediction is what converts a useless teacher into a usable training signal — and it stops mattering once the teacher is good.

Do cluster ensembles help?

Word error rate on dev-other, same reduced setup



- Product k-means splits the 117-dimensional spliced MFCC vector into its zeroth, first and second-order derivative subspaces and quantises each to 100 entries.
- Any single subspace alone is no better than plain k-means, and the zeroth-order one is worse.
- All three together give the best result of any teacher configuration in this experiment.
- The gains are real but modest — roughly one point of WER — and ensembles are not used in the main models.

Worth noticing what is not here: the headline HuBERT models use a single k-means teacher per iteration. Ensembles are presented as a possibility rather than adopted.

What else matters

Masking rate, batch size, and training length

Masking probability

The proportion of frames selected as mask starts is optimal at p = 8%.

Too little masking makes the task trivial; too much removes the context needed to solve it.

Batch size

Increasing the effective batch size, by adding GPUs, significantly improves performance.

Consistent with what was already known from BERT-style training in text.

Training length

Longer training helps consistently. With a 100-cluster MFCC teacher, dev-other WER falls from 17.86 at 100k steps to 11.68 at 800k.

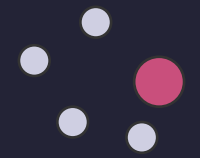
The most striking number on this slide

That 11.68 was obtained with the original, unrefined MFCC teacher — simply by training eight times longer. Compare Discrete BERT at 26.6 in the same comparison, which quantises the same MFCC features into 13,500 units and uses them as both input and target.

So a hundred crude clusters, trained long enough with a masked-only loss, beat thirteen thousand finer ones used the conventional way by a factor of more than two.

5 Appraisal and legacy

What HuBERT enabled, and what it left open



Critical appraisal

What the design buys, and what it costs

Strengths

A plain cross-entropy loss replaces contrastive learning with its negative sampling, diversity term and temperature schedule. Fewer knobs, more stable training.

The architecture is held identical to wav2vec 2.0, so the comparison isolates the objective.

The analyses are unusually good: three interpretable teacher-quality metrics, a stability study, a layer-wise study, and an ablation that identifies a mechanism rather than just a winner.

Scales cleanly to a billion parameters, which several contemporaneous methods did not.

Code and checkpoints released, which is why almost everything downstream builds on it.

Limitations

Not a single training phase. Clustering must be run between iterations, and each iteration is a full pre-training run — the authors name single-phase training as their main future goal.

The layer to cluster from is chosen empirically, and it moves between iterations. There is no principled way to pick it in advance.

Compute-heavy: 256 GPUs for the largest run, with three effective iterations behind it.

Evaluated entirely on speech recognition — the paper flags going beyond ASR as future work.

All audiobook English, so nothing here tests robustness to noise, overlap or domain shift.

What HuBERT enabled

Why it became the default backbone

The SUPERB benchmark	When SUPERB evaluated frozen SSL models across fifteen speech tasks, HuBERT had the best overall generalisation — a result the paper itself did not claim and did not test.
Discrete speech units	HuBERT cluster labels became the standard tokenisation for textless language modelling, speech resynthesis, and the acoustic vocabularies of later speech language models.
WavLM and UniSpeech-SAT	Both extend the HuBERT framework directly rather than wav2vec 2.0's — the second half of this session is one of them.
Multilingual and domain variants	mHuBERT and numerous language-specific reproductions, because the recipe transfers without needing contrastive-loss tuning.
The affective computing connection	Emotion-recognition work that had been building on wav2vec 2.0 largely migrated to HuBERT features — usually with a learned weighted sum across layers, for the reasons we discussed last session.

Take-aways

Four things to carry into the second half

1	**Consistency beats correctness**	A teacher that removes only a quarter of the uncertainty about phone identity is enough, provided its errors are systematic rather than random.
2	**Where the loss is applied is the design**	Masked-only prediction turns a 96% word error rate into 17.9% with the same labels. It is a defence against label noise, and it stops paying once the labels are good.
3	**Bootstrapping works**	Cluster, train, re-cluster on the model's own representations. Two rounds take PNMI from 0.25 to 0.70 with no supervision anywhere in the loop.
4	**Simpler objectives win in the long run**	The contrastive machinery wav2vec 2.0 needed was replaced by a cross-entropy loss, and the field followed.

Questions for discussion

1. The paper argues consistency matters more than correctness. What would a maximally inconsistent teacher look like, and is there a way to measure teacher consistency directly rather than inferring it from downstream word error rate?

2. Masked-only prediction helps with bad targets and hurts with good ones. Given that, should α be annealed across iterations as the teacher improves — and why do you think nobody reports having tried?

3. The layer to cluster from moves from 6 to 9 between iterations, chosen empirically. Is there a principled criterion, and would PNMI across layers be enough to pick it without any labels at all?

4. HuBERT is evaluated entirely on ASR, yet became the default backbone for speaker and paralinguistic tasks. What does that say about how we should evaluate representation-learning papers?

5. Everything here is bootstrapped from k-means on MFCCs — a 1980s feature and a 1950s algorithm. What does it mean that the starting point can be this weak?

Hsu, Bolte, Tsai, Lakhotia, Salakhutdinov & Mohamed (2021), IEEE/ACM TASLP 29, 3451–3460 · arXiv:2106.07447