Overview of Self-Supervised Learning

Junjie LI

Department of Electrical and Electronic Engineering

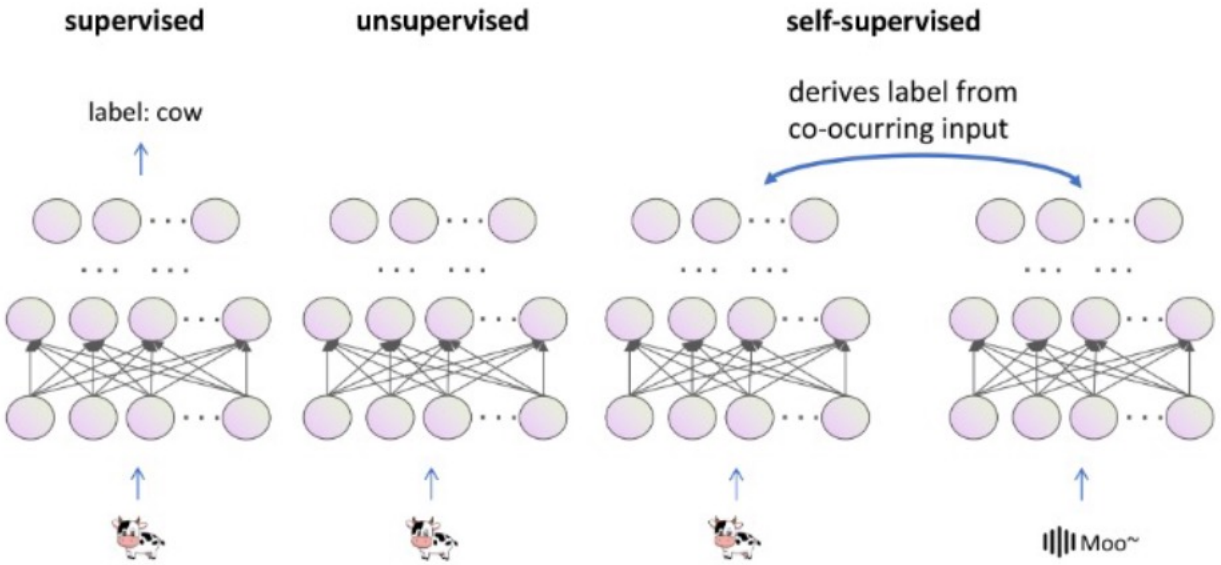
Apr 2025

THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學

Background

What is Self-supervised learning (SSL)?

- Self-supervised learning is a form of unsupervised learning.
- In contrast to supervised learning using human-annotated labels, self-supervised learnings aim to learn the relationship between original data

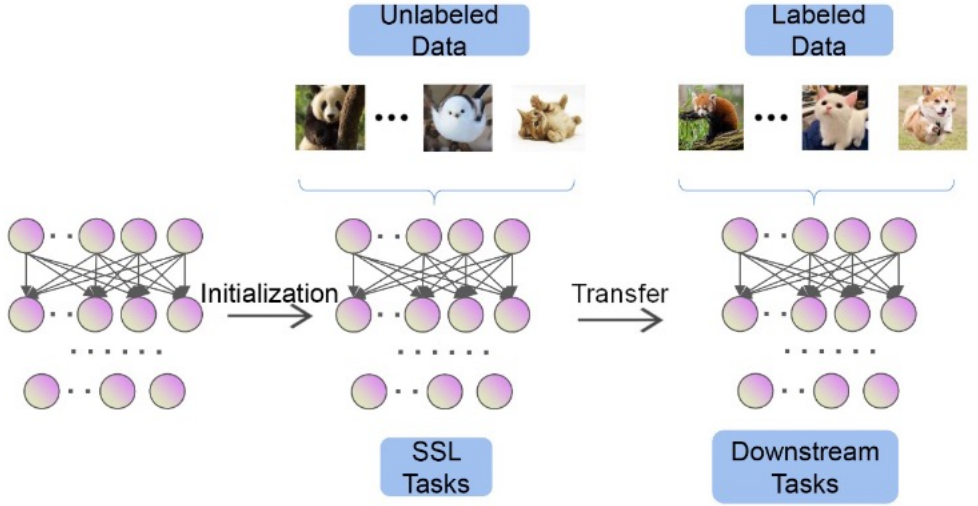


K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in IEEE Conf. Comput. Vis. Pattern Recog., 2020, pp. 9729–9738.

Background

How to use SSL models?

- SSL serves as a kind of pre-training method, after pre-training it could be applied to any downstream tasks.
- Only with a small labeled data to finetune the models, it could get a very good performance.



J. Gui, T. Chen, J. Zhang, Q. Cao, Z. Sun, H. Luo, and D. Tao, "A survey on self-supervised learning: Algorithms, applications, and future trends," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

SSL Training Paradigms

Pretext tasks

- The term “pretext” implies that this task is not our primary objective but rather serves as a means to generate a well-performing and robust pre-trained model.
- In pretext tasks, models are trained with pseudo-labels that are derived from the intrinsic attributes of the data.
- There are a lot of pretext tasks, but not every could contribute to down-stream tasks effectively.
- Next, we summarize some pretext techniques into three categories.

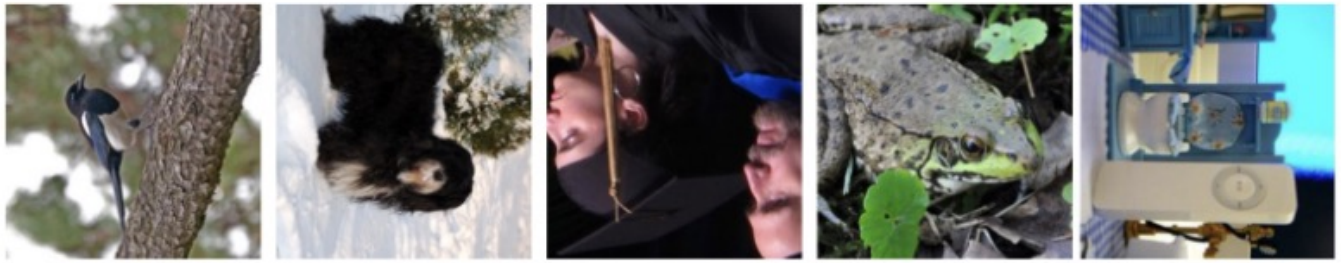
J. Gui, T. Chen, J. Zhang, Q. Cao, Z. Sun, H. Luo, and D. Tao, “A survey on self-supervised learning: Algorithms, applications, and future trends,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

SSL Training Paradigms

Context-Based Methods

- This method rely on the inherent contextual relationships in the data, such as spatial information, color information and semantic information.

- For example, Gidaris et al. trained a network to learn visual representations with rotation degrees as target.

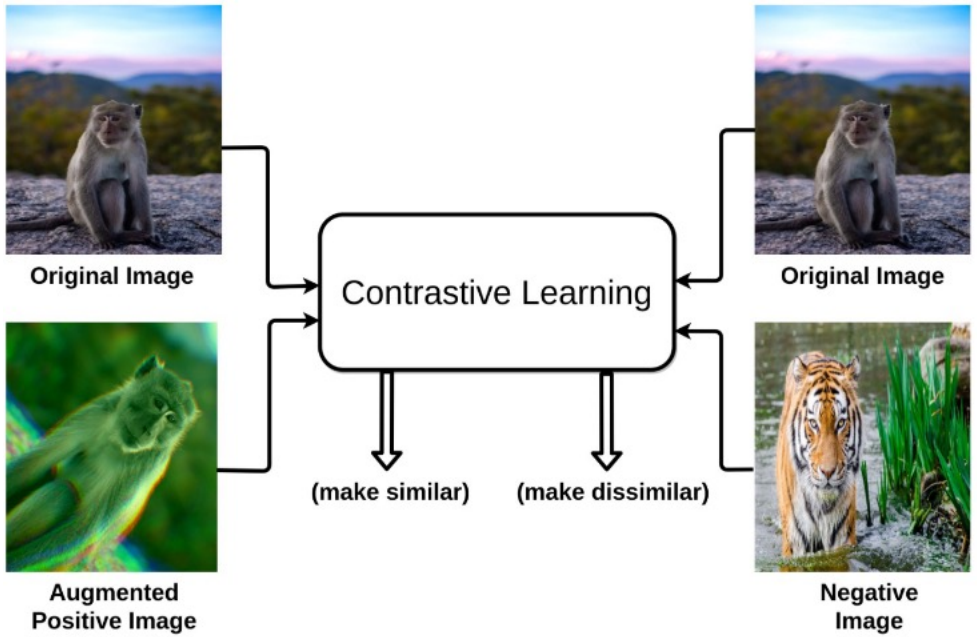


S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” arXiv preprint arXiv:1803.07728, 2018

SSL Training Paradigms

Contrastive Learning

- It is a discriminative method that aims to minimize the distance between similar samples while maximizing the distance between dissimilar ones.

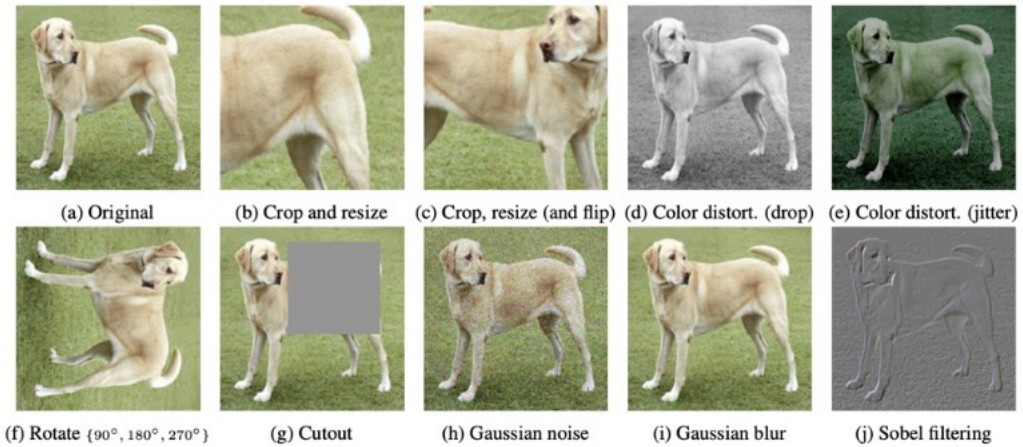


A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," Technologies, vol. 9, no. 1, p. 2, 2020.

SSL Training Paradigms

Contrastive Learning

- Augmentation methods



- Loss functions of infoNCE

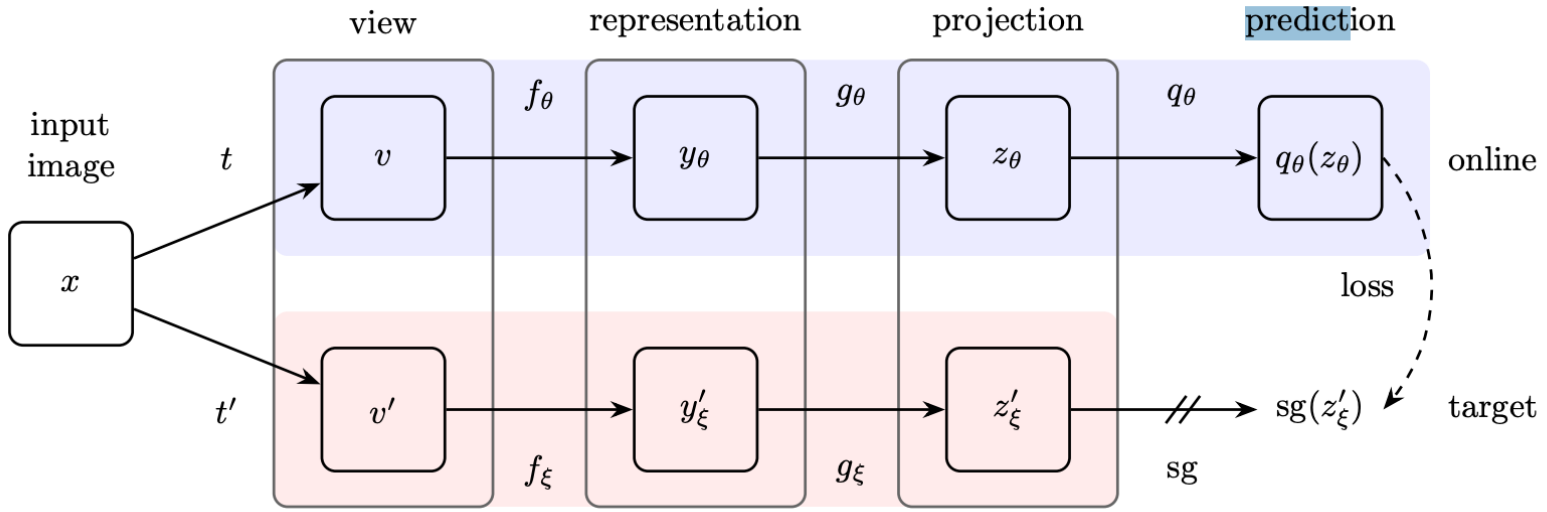
$$L_{\text{infoNCE}} = -\log \frac{\exp(\text{sim}(q, k_+)/\tau)}{\exp(\text{sim}(q, k_+)/\tau) + \sum_i^K \exp(\text{sim}(q, k_i)/\tau)}. \quad (5)$$

A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," Technologies, vol. 9, no. 1, p. 2, 2020.

SSL Training Paradigms

Contrastive Learning

- **Non-negative methods:** Some studies explore to eliminate the need for negative pairs.



$$\mathcal{L}_{\theta,\xi} \triangleq \left\| \overline{q_\theta}(z_\theta) - \overline{z}'_\xi \right\|_2^2 = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\left\| q_\theta(z_\theta) \right\|_2 \cdot \left\| z'_\xi \right\|_2}.$$

- BYOL utilizes two identical encoders, known as Siamese networks, with the same architecture but different weights.
- One branch is updated online by gradient, the other one updates its parameters through a slow-moving average

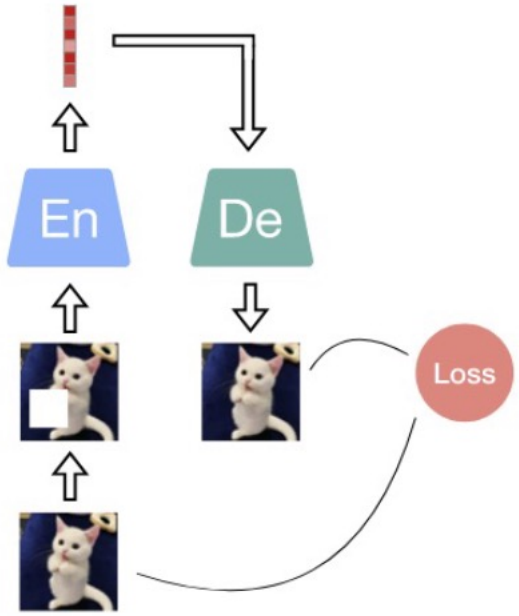
$$\xi \leftarrow \tau\xi + (1-\tau)\theta,$$

J.-B. Grill, F. Strub, F. Altch´e, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar et al., "Bootstrap your own latent-a new approach to self-supervised learning," Advances in Neural Information Processing Systems, vol. 33, pp.21 271–21 284, 2020.

SSL Training Paradigms

Generative Algorithms

- Generative methods aim to reconstruct original views from masked or inpainted inputs.

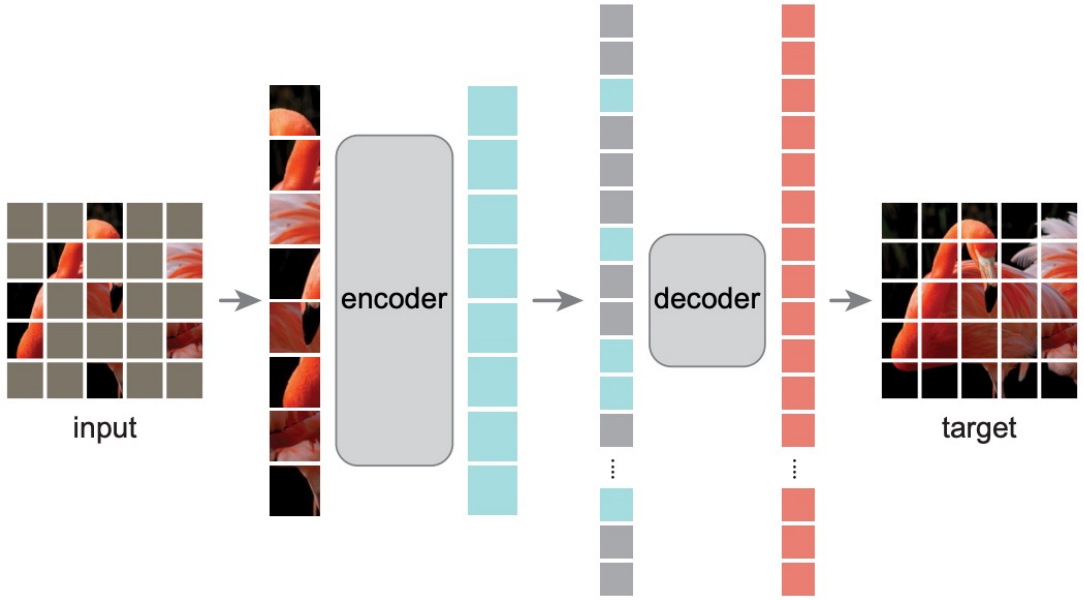


S. Liu, A. Mallol-Ragolta, E. Parada-Cabaleiro, K. Qian, X. Jing, A. Kathan, B. Hu, and B. W. Schuller, "Audio self-supervised learning: A survey," Patterns, vol. 3, no. 12, 2022.

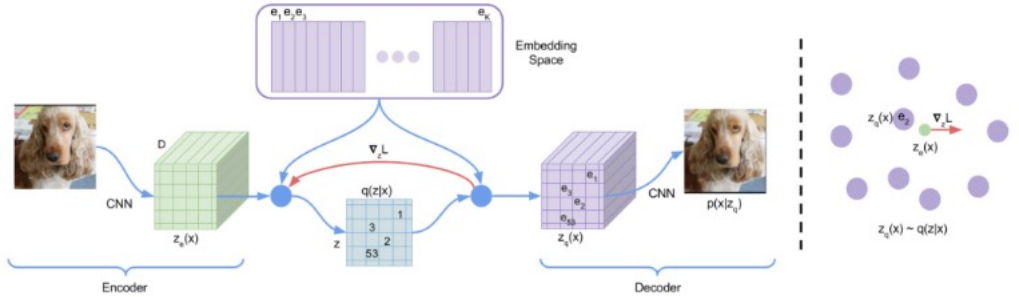
SSL Training Paradigms

Generative Algorithms

- MAE



- VQ-VAE



- K. He, X. Chen, S. Xie, Y. Li, P. Doll´ar, and R. Girshick, “Masked autoencoders are scalable vision learners,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 16 000–16 009.
- A. Van Den Oord, O. Vinyals et al., “Neural discrete representation learning,” Advances in neural information processing systems, vol. 30, 2017.

Applications in Speech

The intrinsic characteristics of speech

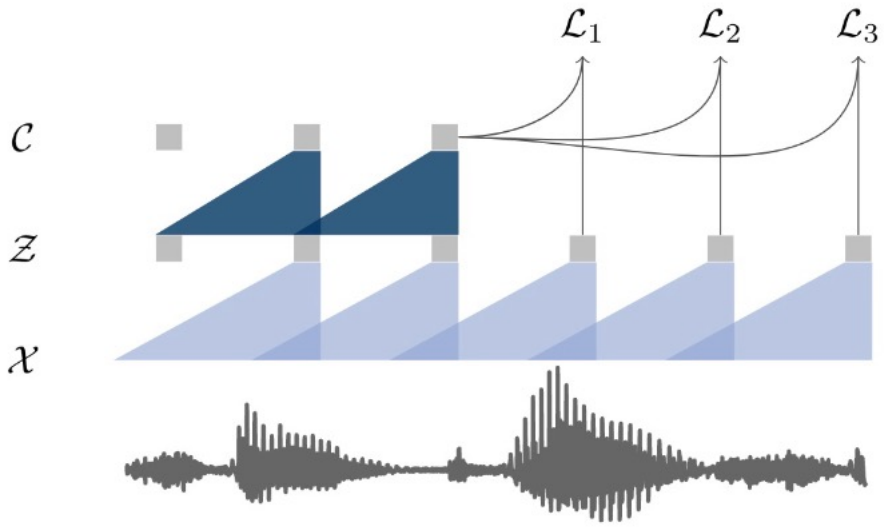
- Speech is a sequence and doesn't have a fixed length
- The samplling rate of speech is very high which causes the length of speech is long
- Speech is continuous. Text could be broken into words, or subwords, but speech is hard to do the same thing.
- Speech processing tasks are diverse. Cause speech contain much more useful information than text, developing one SSL model for all tasks is very difficult

A. Mohamed, H.-y. Lee, L. Borgholt, J. D. Havtorn, J. Edin, C. Igel, K. Kirchhoff, S.-W. Li, K. Livescu, L. Maaløe et al., "Self-supervised speech representation learning: A review," IEEE Journal of Selected Topics in Signal Processing, vol. 16, no. 6, pp. 1179–1210, 2022.

Applications in Speech

wav2vec

- The encoder network embeds the audio signal in a latent space, $f: \mathcal{X} \rightarrow \mathcal{Z}$.
- The context network aims to mix multiple latent representations into a single contextualized tensor: g: $\mathcal{Z} \rightarrow \mathcal{C}$
- Both encoder and context networks are convolutions layers.



S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," arXiv preprint arXiv:1904.05862, 2019.

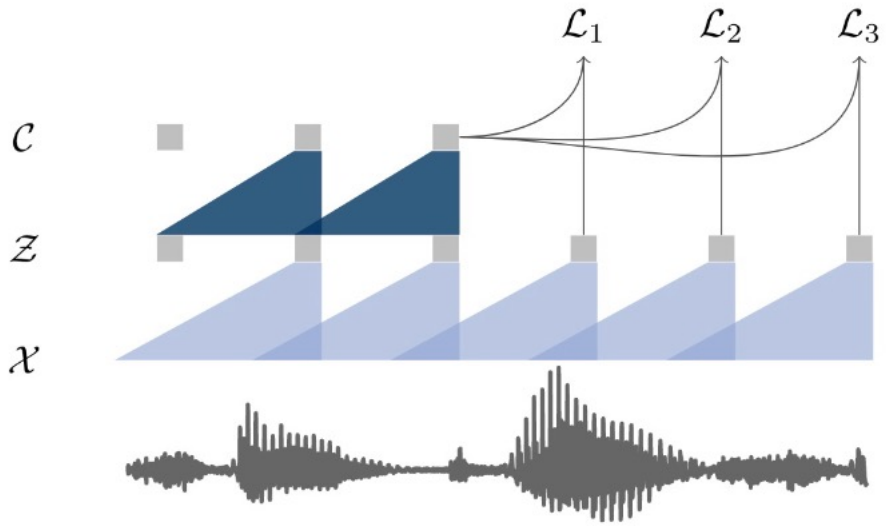
Applications in Speech

wav2vec

- The wav2vec predicts future information based on previous inputs.
- Contrastive loss

$$\mathcal{L}_k = -\sum_{i=1}^{T-k}\left(\log\sigma(\mathbf{z}_{i+k}^{\top}h_k(\mathbf{c}_i)) + \lambda \mathop{\mathbb{E}}_{\tilde{\mathbf{z}}\sim p_n}[\log\sigma(-\tilde{\mathbf{z}}^{\top}h_k(\mathbf{c}_i))]\right), \tag{9}$$

Anchor: $h_k(c_i)$
Positive samples: z_{i+k}
Negative samples: $\tilde{z}$

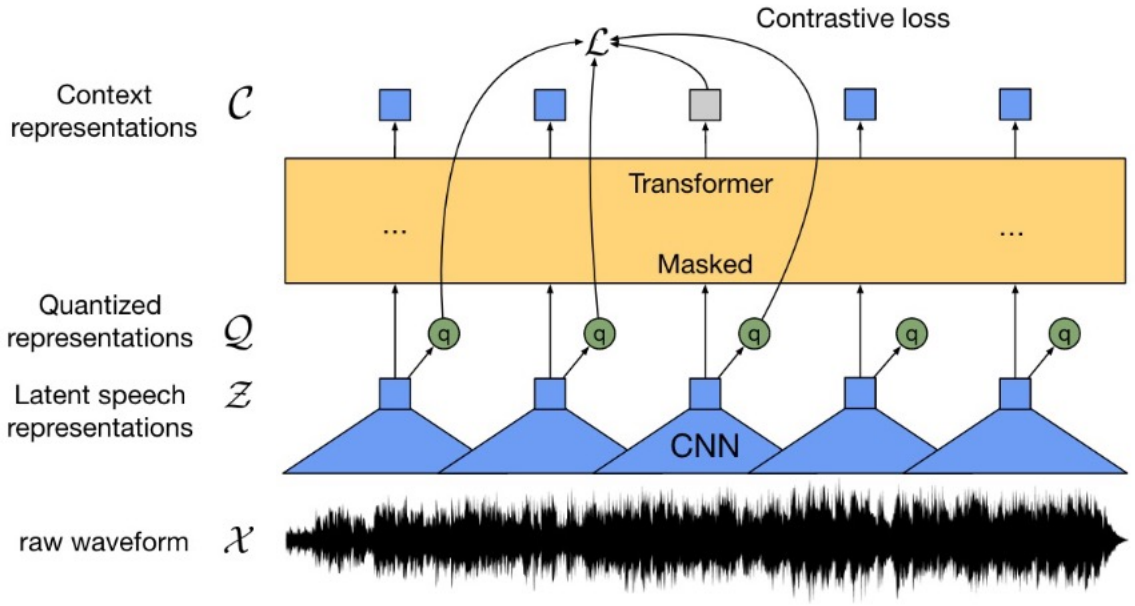


S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," arXiv preprint arXiv:1904.05862, 2019.

Applications in Speech

wav2vec 2.0

- Compared to wav2vec, the wav2vec2.0 makes some changes:
  1. Replacing original convolution context network with Transformer module.
  2. Adding a quantization operation.
  3. Masking some parts of latent speech representations.
  4. Instead of only predicting future information, wav2vec 2.0 utilizes both historical and future information.



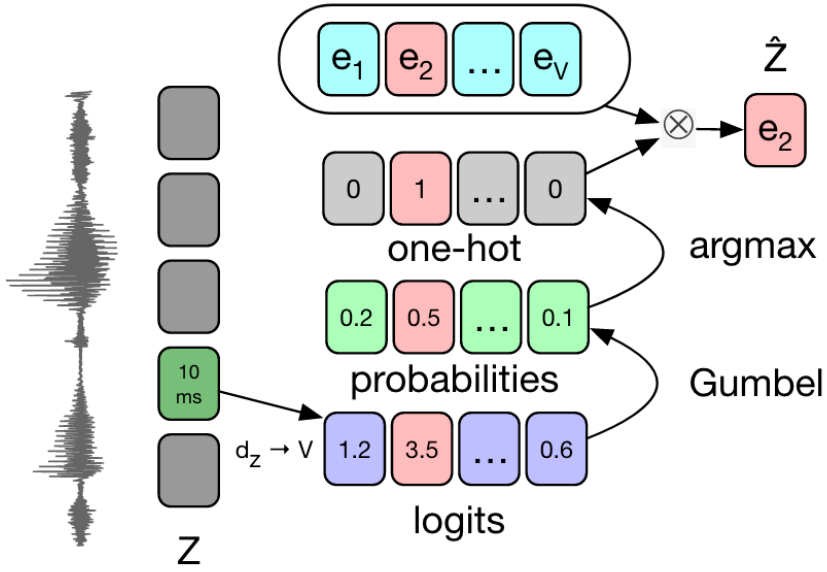
A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” Advances in neural information processing systems, vol. 33, pp. 12 449-12 460, 2020.

Applications in Speech

Wav2vec 2.0

- Quantization operation (Gumbel softmax)

  1. Transforming latent speech representations into logits with ReLU and a linear layer.
  2. Transforming logitis to probabilities.
  3. Selecting the codebook entry with highest probability.



(a) Gumbel-Softmax

A. Van Den Oord, O. Vinyals et al., "Neural discrete representation learning," Advances in neural information processing systems, vol. 30, 2017

Applications in Speech

Wav2vec 2.0

- Loss function $\mathcal{L} = \mathcal{L}_m + \alpha \mathcal{L}_d$

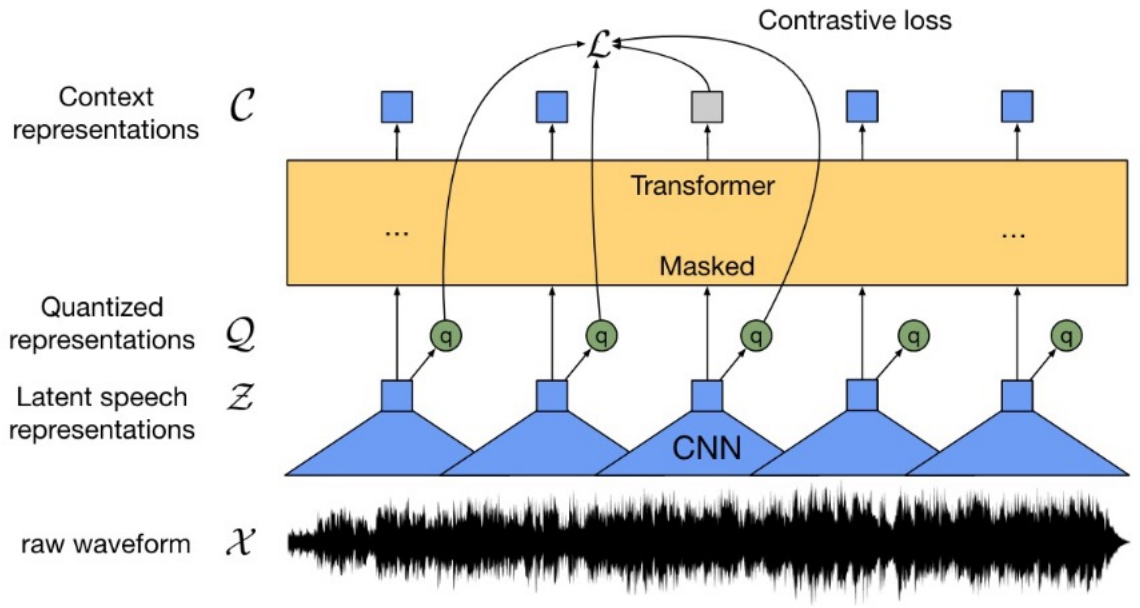
1. Contrastive loss

$$\mathcal{L}_m = -\log \frac{\exp(sim(\mathbf{c}_t, \mathbf{q}_t)/\kappa)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp(sim(\mathbf{c}_t, \tilde{\mathbf{q}})/\kappa)}$$

2. Diversity loss

$$\mathcal{L}_d = \frac{1}{GV} \sum_{g=1}^{G} -H(\bar{p}_g) = \frac{1}{GV} \sum_{g=1}^{G} \sum_{v=1}^{V} \bar{p}_{g,v} \log \bar{p}_{g,v}$$

$$p_{g,v} = \frac{\exp(l_{g,v} + n_v)/\tau}{\sum_{k=1}^{V} \exp(l_{g,k} + n_k)/\tau},$$

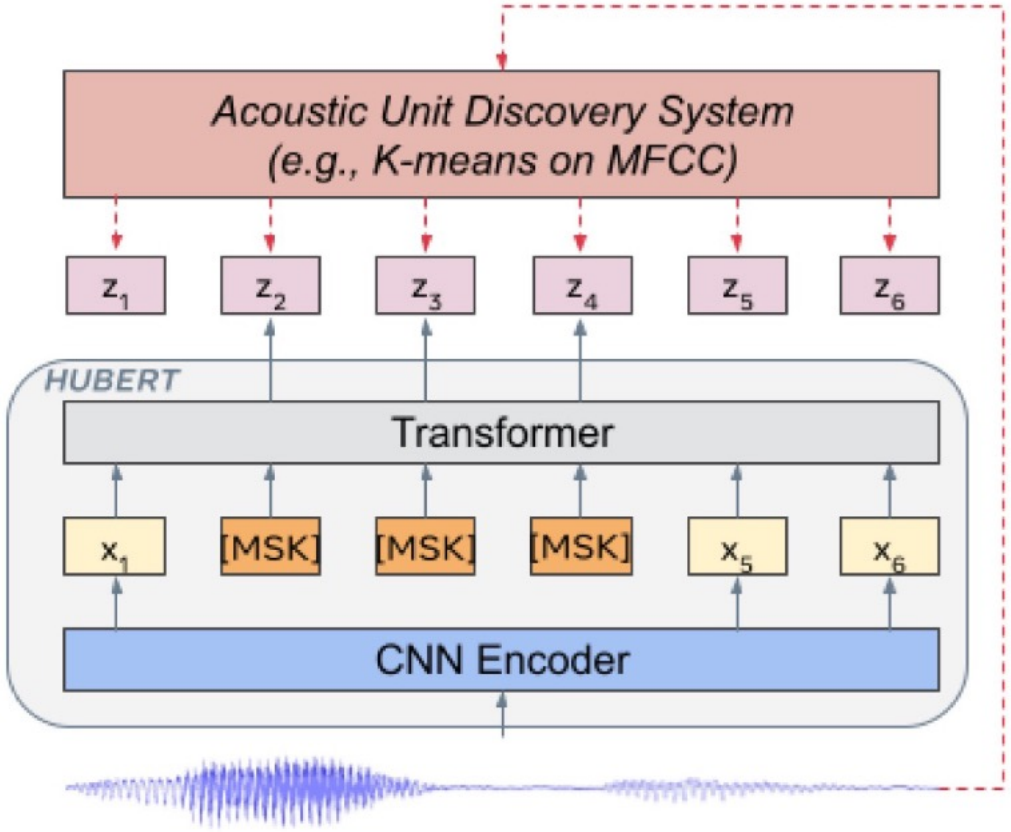


A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," Advances in neural information processing systems, vol. 33, pp. 12 449-12 460, 2020.

Applications in Speech

Hubert

- The structure of Hubert is similar to wav2vec 2.0.
  1. CNN encoder to extract speech feature
  2. Masking some speech features
  3. Transformer aims to learn contectual information.
- Some differences:
  1. No quatilizations
  2. Using k-means to get pseduo-classification lables.
  3. Not using contrastive loss



W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," IEEE Trans. on Audio, Speech, and Language Processing, vol. 29, pp. 3451–3460, 2021.

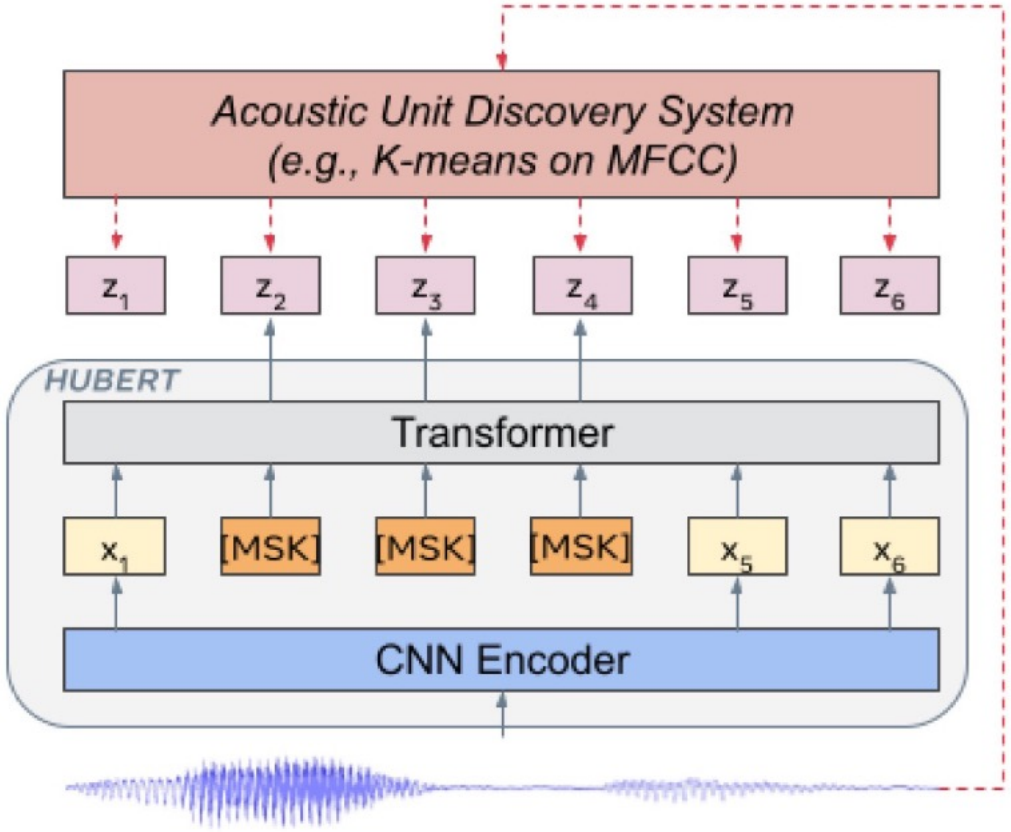
Applications in Speech

Hubert

- Training
  1. In the first iterative, applying k-means on MFCC to get classification labels.
  2. In the next iteratives, applying k-means on hidden units in the last iterative.

- Cross-entropy loss

$$\mathcal{L}_m = \sum_{t \in \mathcal{M}} -\log p(c_t \mid X) \ ,$$
$$\mathcal{L} = \beta \mathcal{L}_m + (1-\beta)\mathcal{L}_u \ .$$



W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," IEEE Trans. on Audio, Speech, and Language Processing, vol. 29, pp. 3451–3460, 2021.

Applications in Speech

WavLM

- WavLM is quite similar to Hubert.
- But it emphasizes spoken content modeling and speaker identity preservation.
  1. First, adding a gated relative position bias into transformer.
  2. Second, the inputs to WavLM contain some mixture data, which forces model to learn to keep target speech and filter out noisy signals.

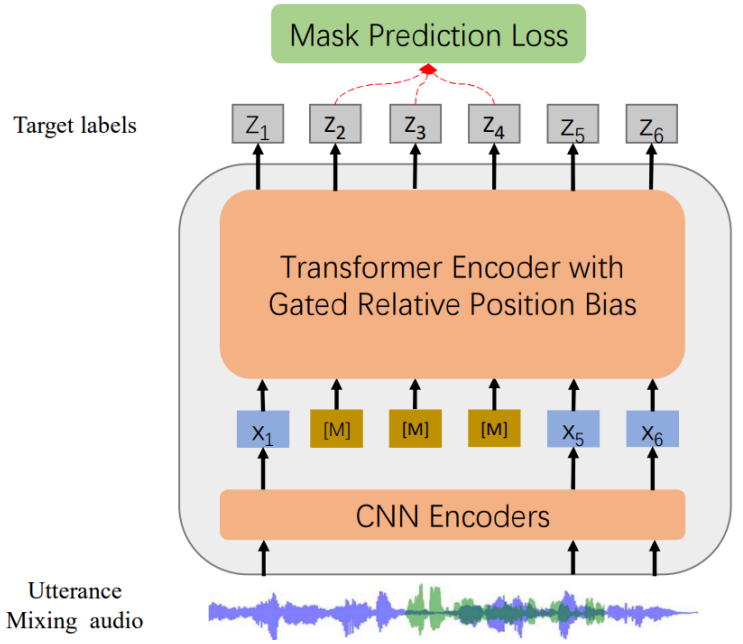


Fig. 1. Model Architecture.

S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao et al., "Wavlm: Large-scale self-supervised pre- training for full stack speech processing," IEEE Journal of Selected Topics in Signal Processing, vol. 16, no. 6, pp. 1505–1518, 2022.

Downstream Tasks in Speech

- Content
  - Phoneme Recognition (PR)
  - Automatic Speech Recognition(ASR)
  - Keyword Spotting (KS)
  - Query by Example Spoken Term Detection (QbE)
- Speaker
  - Speaker Identification (SID)
  - Automatic Speaker Verification (ASV)
  - Speaker Diarization (SD)
- Semantics
  - Speech Translation (ST)
  - Intent Classification (IC)
  - Slot Filling (SF)
- Paralinguistics
  - Emotion Recognition (ER)
- Generation
  - Speech Enhancement (SE)
  - Speech Separation (SS)
  - Voice Conversation (VC)

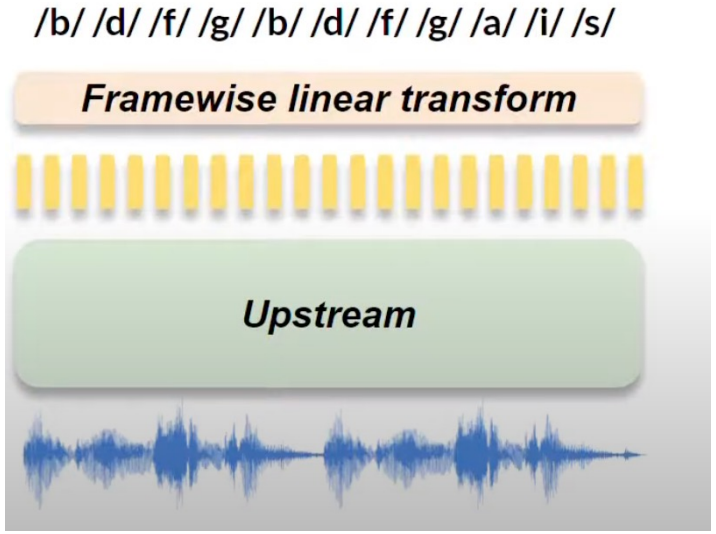
S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin et al., "Superb: Speech processing universal performance benchmark," arXiv preprint arXiv:2105.01051, 2021.

Experiments

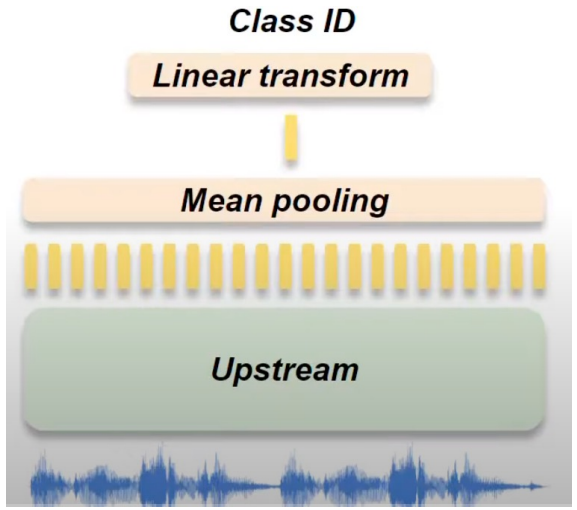
Following setups in SUPERB:

- Pre-trained SSL models are frozen.

- **PR, KS, SID,IC, ER** are simple tasks that are solvable with linear downstream models.



PR



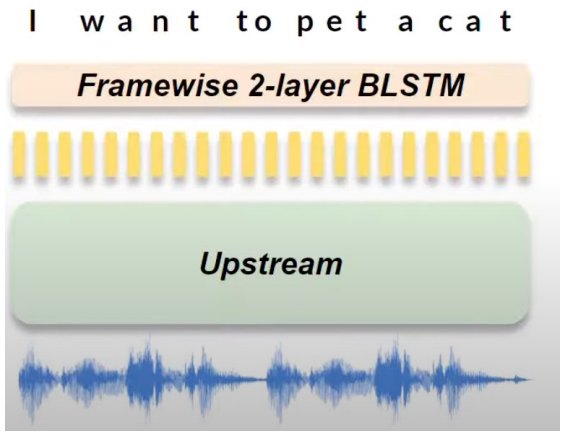
KS/SID/IC/ER

S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin et al., "Superb: Speech processing universal performance benchmark," arXiv preprint arXiv:2105.01051, 2021.

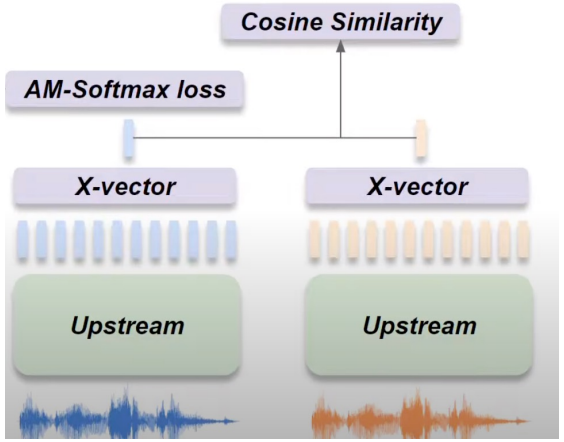
Experiments

Following setups in SUPERB:

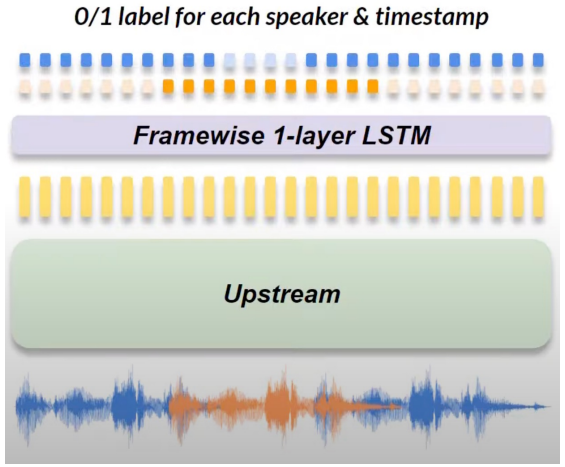
- For other tasks, downstream models are simple non-linear layers.



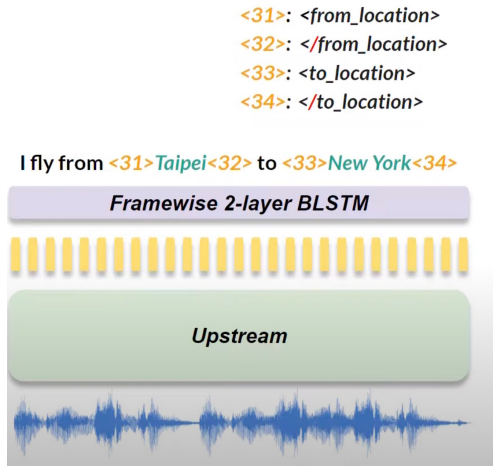
ASR



ASV



SD



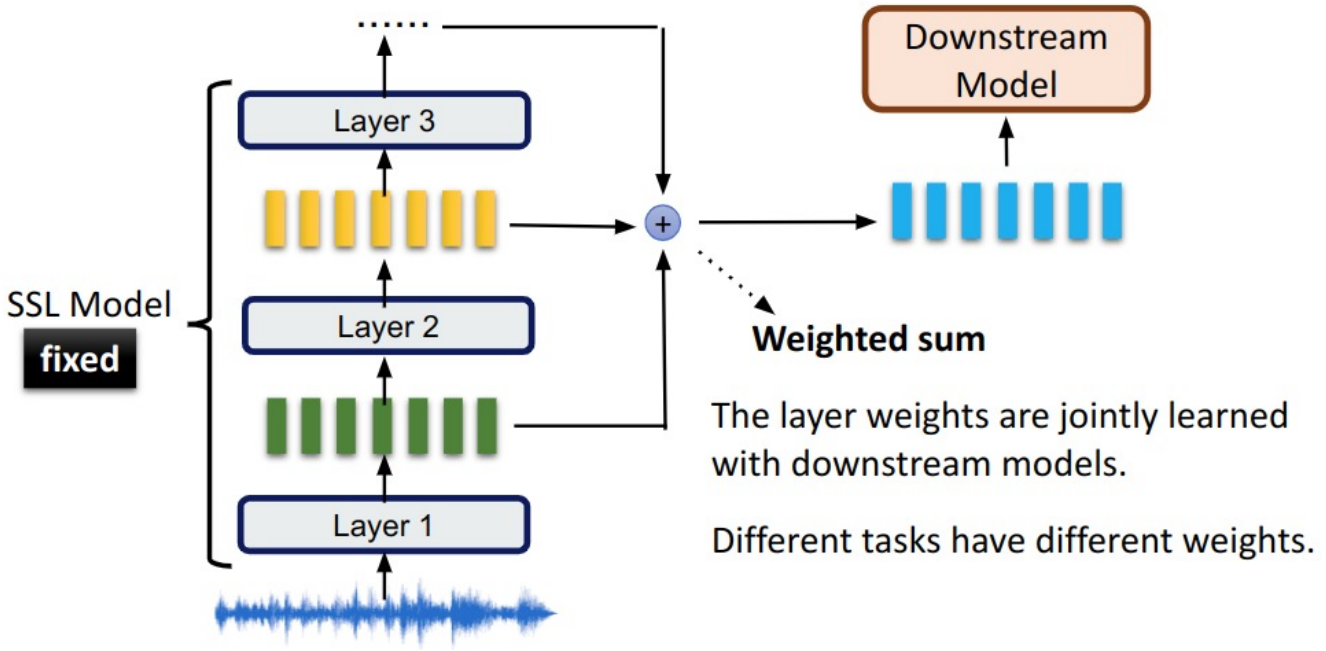
SF

S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin et al., "Superb: Speech processing universal performance benchmark," arXiv preprint arXiv:2105.01051, 2021.

Experiments

Following setups in SUPERB:

- The downstream models consume the weighted sum results of the hidden states extracted from each layer of the pretrained model.



S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin et al., "Superb: Speech processing universal performance benchmark," arXiv preprint arXiv:2105.01051, 2021.

Results

Comparation within SSL models

Method	#Params	Corpus	Speaker			Content					Semantics				ParaL	Generation						Overall
			SID	ASV	SD	PR	ASR	OOD-ASR	KS	QbE	ST	IC	SF		ER	SE		SS	VC			
			Acc ↑	EER ↓	DER ↓	PER ↓	WER ↓	WER ↓	Acc ↑	MTWV ↑	BLEU ↑	Acc ↑	F1 ↑	CER ↓	Acc ↑	PESQ ↑	STOI ↑	SI-SDRi ↑	MCD ↓	WER ↓	ASV ↑	Score ↑
FBANK	0	-	8.5E-4	9.56	10.05	82.01	23.18	63.58	8.63	0.0058	2.32	9.10	69.64	52.94	35.39	2.55	93.6	9.23	8.47	38.3	77.25	43.2
PASE+ [44]	7.83M	LS 50 hr	37.99	11.61	8.68	58.87	25.11	61.56	82.54	0.0072	3.16	29.82	62.14	60.17	57.86	2.56	93.9	9.87	8.66	30.6	63.20	51.5
APC [30]	4.11M	LS 360 hr	60.42	8.56	10.53	41.98	21.28	63.12	91.01	0.0310	5.95	74.69	70.46	50.89	59.33	2.56	93.4	8.92	8.05	27.2	87.25	59.2
VQ-APC [29]	4.63M	LS 360 hr	60.15	8.72	10.45	41.08	21.20	63.56	91.11	0.0251	4.23	74.48	68.53	52.91	59.66	2.56	93.4	8.44	7.84	22.4	94.25	59.5
NPC [33]	19.38M	LS 360 hr	55.92	9.40	9.34	43.81	20.20	61.66	88.96	0.0246	4.32	69.44	72.79	48.44	59.08	2.52	93.1	8.04	7.86	30.4	94.75	59.0
Mockingjay [35]	85.12M	LS 360 hr	32.29	11.66	10.54	70.19	22.82	65.27	83.67	6.6E-04	4.45	34.33	61.59	58.89	50.28	2.53	93.4	9.29	8.29	35.1	79.75	51.0
TERA [34]	21.33M	LS 960 hr	57.57	15.89	9.96	49.17	18.17	58.49	89.48	0.0013	5.66	58.42	67.50	54.17	56.27	2.54	93.6	10.19	8.21	25.1	83.75	57.2
DeCoAR 2.0 [37]	89.84M	LS 960 hr	74.42	7.16	6.59	14.93	13.02	53.62	94.48	0.0406	9.94	90.80	83.28	34.73	62.47	2.47	93.2	8.54	7.83	17.1	90.75	66.3
modified CPC [53]	1.84M	LL 60k hr	39.63	12.86	10.38	42.54	20.18	62.54	91.88	0.0326	4.82	64.09	71.19	49.91	60.96	2.57	93.7	10.40	8.41	26.2	71.00	56.9
wav2vec [39]	32.54M	LS 960 hr	56.56	7.99	9.9	31.58	15.86	55.86	95.59	0.0485	6.61	84.92	76.37	43.71	59.79	2.53	93.8	9.30	7.45	10.1	98.25	63.5
vq-wav2vec [40]	34.15M	LS 960 hr	38.80	10.38	9.93	33.48	17.71	60.66	93.38	0.0410	5.66	85.68	77.68	41.54	58.24	2.48	93.6	8.16	**7.08**	13.4	**100.00**	61.8
wav2vec 2.0 Base [5]	95.04M	LS 960 hr	75.18	6.02	6.08	5.74	6.43	46.95	96.23	0.0233	14.81	92.35	88.30	24.77	63.43	2.55	93.9	9.77	7.50	10.5	98.00	69.6
HuBERT Base [6]	94.68M	LS 960 hr	81.42	5.11	5.88	5.41	6.42	46.69	96.30	0.0736	15.53	98.34	88.53	25.20	64.92	2.58	93.9	9.36	7.47	8.0	98.50	70.9
WavLM Base	94.70M	LS 960 hr	84.51	4.69	4.55	4.84	6.21	42.81	96.79	0.0870	20.74	98.63	89.38	22.86	65.94	2.58	94.0	10.37	7.42	8	98.00	72.0
- w/o denoising task	94.70M	LS 960 hr	84.39	4.91	6.03	4.85	6.08	43.61	96.79	0.0799	21.03	98.42	88.69	23.43	65.55	2.56	93.9	9.91	7.43	**7.5**	97.75	71.7
- w/o structure modification	94.68M	LS 960 hr	84.74	4.61	4.72	5.22	6.80	42.88	96.79	0.0956	20.03	98.31	88.56	24.00	65.60	2.58	94.0	10.29	7.45	8.4	99.00	71.9
WavLM Base+	94.70M	Mix 94k hr	89.42	4.07	3.50	3.92	5.59	38.32	97.37	**0.0988**	24.25	99.00	90.58	21.20	68.65	2.63	94.3	10.85	7.40	8.1	99.00	73.4
wav2vec 2.0 Large [5]	317.38M	LL 60k hr	86.14	5.65	5.62	4.75	3.75	44.69	96.66	0.0489	12.48	95.28	87.11	27.31	65.64	2.52	94.0	10.02	7.63	15.8	97.25	70.4
HuBERT Large [6]	316.61M	LL 60k hr	90.33	5.98	5.75	3.53	3.62	44.08	95.29	0.0353	20.01	98.76	89.81	21.76	67.62	2.64	94.2	10.45	7.22	9.0	99.25	72.2
WavLM Large	316.62M	Mix 94k hr	**95.49**	**3.77**	**3.24**	**3.06**	**3.44**	**32.27**	**97.86**	0.0886	**26.57**	**99.31**	**92.21**	**18.36**	**70.62**	**2.70**	**94.5**	**11.19**	7.30	9.0	99.00	**74.6**

1. SSL features are better than FBANK feature in many tasks, especially on tasks with one linear layer modeling, like **PR, KS, SID,IC, ER**

S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin et al., "Superb: Speech processing universal performance benchmark," arXiv preprint arXiv:2105.01051, 2021.

S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao et al., "Wavlm: Large-scale self-supervised pre- training for full stack speech processing," IEEE Journal of Selected Topics in Signal Processing, vol. 16, no. 6, pp. 1505–1518, 2022.

Results

Comparation within SSL models

Method	#Params	Corpus	Speaker			Content					Semantics				ParaL	Generation						Overall
			SID	ASV	SD	PR	ASR	OOD-ASR	KS	QbE	ST	IC	SF		ER	SE		SS	VC			
			Acc ↑	EER ↓	DER ↓	PER ↓	WER ↓	WER ↓	Acc ↑	MTWV ↑	BLEU ↑	Acc ↑	F1 ↑	CER ↓	Acc ↑	PESQ ↑	STOI ↑	SI-SDRi ↑	MCD ↓	WER ↓	ASV ↑	Score ↑
FBANK	0	-	8.5E-4	9.56	10.05	82.01	23.18	63.58	8.63	0.0058	2.32	9.10	69.64	52.94	35.39	2.55	93.6	9.23	8.47	38.3	77.25	43.2
PASE+ [44]	7.83M	LS 50 hr	37.99	11.61	8.68	58.87	25.11	61.56	82.54	0.0072	3.16	29.82	62.14	60.17	57.86	2.56	93.9	9.87	8.66	30.6	63.20	51.5
APC [30]	4.11M	LS 360 hr	60.42	8.56	10.53	41.98	21.28	63.12	91.01	0.0310	5.95	74.69	70.46	50.89	59.33	2.56	93.4	8.92	8.05	27.2	87.25	59.2
VQ-APC [29]	4.63M	LS 360 hr	60.15	8.72	10.45	41.08	21.20	63.56	91.11	0.0251	4.23	74.48	68.53	52.91	59.66	2.56	93.4	8.44	7.84	22.4	94.25	59.5
NPC [33]	19.38M	LS 360 hr	55.92	9.40	9.34	43.81	20.20	61.66	88.96	0.0246	4.32	69.44	72.79	48.44	59.08	2.52	93.1	8.04	7.86	30.4	94.75	59.0
Mockingjay [35]	85.12M	LS 360 hr	32.29	11.66	10.54	70.19	22.82	65.27	83.67	6.6E-04	4.45	34.33	61.59	58.89	50.28	2.53	93.4	9.29	8.29	35.1	79.75	51.0
TERA [34]	21.33M	LS 960 hr	57.57	15.89	9.96	49.17	18.17	58.49	89.48	0.0013	5.66	58.42	67.50	54.17	56.27	2.54	93.6	10.19	8.21	25.1	83.75	57.2
DeCoAR 2.0 [37]	89.84M	LS 960 hr	74.42	7.16	6.59	14.93	13.02	53.62	94.48	0.0406	9.94	90.80	83.28	34.73	62.47	2.47	93.2	8.54	7.83	17.1	90.75	66.3
modified CPC [53]	1.84M	LL 60k hr	39.63	12.86	10.38	42.54	20.18	62.54	91.88	0.0326	4.82	64.09	71.19	49.91	60.96	2.57	93.7	10.40	8.41	26.2	71.00	56.9
wav2vec [39]	32.54M	LS 960 hr	56.56	7.99	9.9	31.58	15.86	55.86	95.59	0.0485	6.61	84.92	76.37	43.71	59.79	2.53	93.8	9.30	7.45	10.1	98.25	63.5
vq-wav2vec [40]	34.15M	LS 960 hr	38.80	10.38	9.93	33.48	17.71	60.66	93.38	0.0410	5.66	85.68	77.68	41.54	58.24	2.48	93.6	8.16	**7.08**	13.4	**100.00**	61.8
wav2vec 2.0 Base [5]	95.04M	LS 960 hr	75.18	6.02	6.08	5.74	6.43	46.95	96.23	0.0233	14.81	92.35	88.30	24.77	63.43	2.55	93.9	9.77	7.50	10.5	98.00	69.6
HuBERT Base [6]	94.68M	LS 960 hr	81.42	5.11	5.88	5.41	6.42	46.69	96.30	0.0736	15.53	98.34	88.53	25.20	64.92	2.58	93.9	9.36	7.47	8.0	98.50	70.9
WavLM Base	94.70M	LS 960 hr	84.51	4.69	4.55	4.84	6.21	42.81	96.79	0.0870	20.74	98.63	89.38	22.86	65.94	2.58	94.0	10.37	7.42	8	98.00	72.0
- w/o denoising task	94.70M	LS 960 hr	84.39	4.91	6.03	4.85	6.08	43.61	96.79	0.0799	21.03	98.42	88.69	23.43	65.55	2.56	93.9	9.91	7.43	**7.5**	97.75	71.7
- w/o structure modification	94.68M	LS 960 hr	84.74	4.61	4.72	5.22	6.80	42.88	96.79	0.0956	20.03	98.31	88.56	24.00	65.60	2.58	94.0	10.29	7.45	8.4	99.00	71.9
WavLM Base+	94.70M	Mix 94k hr	89.42	4.07	3.50	3.92	5.59	38.32	97.37	**0.0988**	24.25	99.00	90.58	21.20	68.65	2.63	94.3	10.85	7.40	8.1	99.00	73.4
wav2vec 2.0 Large [5]	317.38M	LL 60k hr	86.14	5.65	5.62	4.75	3.75	44.69	96.66	0.0489	12.48	95.28	87.11	27.31	65.64	2.52	94.0	10.02	7.63	15.8	97.25	70.4
HuBERT Large [6]	316.61M	LL 60k hr	90.33	5.98	5.75	3.53	3.62	44.08	95.29	0.0353	20.01	98.76	89.81	21.76	67.62	2.64	94.2	10.45	7.22	9.0	99.25	72.2
WavLM Large	316.62M	Mix 94k hr	**95.49**	**3.77**	**3.24**	**3.06**	**3.44**	**32.27**	**97.86**	0.0886	**26.57**	**99.31**	**92.21**	**18.36**	**70.62**	**2.70**	**94.5**	**11.19**	7.30	9.0	99.00	**74.6**

2. There are some exceptionts that FBANK is better.

S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin et al., "Superb: Speech processing universal performance benchmark," arXiv preprint arXiv:2105.01051, 2021.

S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao et al., "Wavlm: Large-scale self-supervised pre- training for full stack speech processing," IEEE Journal of Selected Topics in Signal Processing, vol. 16, no. 6, pp. 1505–1518, 2022.

Results

Comparation within SSL models

Method	#Params	Corpus	Speaker			Content					Semantics				ParaL	Generation						Overall
			SID	ASV	SD	PR	ASR	OOD-ASR	KS	QbE	ST	IC	SF		ER	SE		SS	VC			
			Acc ↑	EER ↓	DER ↓	PER ↓	WER ↓	WER ↓	Acc ↑	MTWV ↑	BLEU ↑	Acc ↑	F1 ↑	CER ↓	Acc ↑	PESQ ↑	STOI ↑	SI-SDRi ↑	MCD ↓	WER ↓	ASV ↑	Score ↑
FBANK	0	-	8.5E-4	9.56	10.05	82.01	23.18	63.58	8.63	0.0058	2.32	9.10	69.64	52.94	35.39	2.55	93.6	9.23	8.47	38.3	77.25	43.2
PASE+ [44]	7.83M	LS 50 hr	37.99	11.61	8.68	58.87	25.11	61.56	82.54	0.0072	3.16	29.82	62.14	60.17	57.86	2.56	93.9	9.87	8.66	30.6	63.20	51.5
APC [30]	4.11M	LS 360 hr	60.42	8.56	10.53	41.98	21.28	63.12	91.01	0.0310	5.95	74.69	70.46	50.89	59.33	2.56	93.4	8.92	8.05	27.2	87.25	59.2
VQ-APC [29]	4.63M	LS 360 hr	60.15	8.72	10.45	41.08	21.20	63.56	91.11	0.0251	4.23	74.48	68.53	52.91	59.66	2.56	93.4	8.44	7.84	22.4	94.25	59.5
NPC [33]	19.38M	LS 360 hr	55.92	9.40	9.34	43.81	20.20	61.66	88.96	0.0246	4.32	69.44	72.79	48.44	59.08	2.52	93.1	8.04	7.86	30.4	94.75	59.0
Mockingjay [35]	85.12M	LS 360 hr	32.29	11.66	10.54	70.19	22.82	65.27	83.67	6.6E-04	4.45	34.33	61.59	58.89	50.28	2.53	93.4	9.29	8.29	35.1	79.75	51.0
TERA [34]	21.33M	LS 960 hr	57.57	15.89	9.96	49.17	18.17	58.49	89.48	0.0013	5.66	58.42	67.50	54.17	56.27	2.54	93.6	10.19	8.21	25.1	83.75	57.2
DeCoAR 2.0 [37]	89.84M	LS 960 hr	74.42	7.16	6.59	14.93	13.02	53.62	94.48	0.0406	9.94	90.80	83.28	34.73	62.47	2.47	93.2	8.54	7.83	17.1	90.75	66.3
modified CPC [53]	1.84M	LL 60k hr	39.63	12.86	10.38	42.54	20.18	62.54	91.88	0.0326	4.82	64.09	71.19	49.91	60.96	2.57	93.7	10.40	8.41	26.2	71.00	56.9
wav2vec [39]	32.54M	LS 960 hr	56.56	7.99	9.9	31.58	15.86	55.86	95.59	0.0485	6.61	84.92	76.37	43.71	59.79	2.53	93.8	9.30	7.45	10.1	98.25	63.5
vq-wav2vec [40]	34.15M	LS 960 hr	38.80	10.38	9.93	33.48	17.71	60.66	93.38	0.0410	5.66	85.68	77.68	41.54	58.24	2.48	93.6	8.16	**7.08**	13.4	**100.00**	61.8
wav2vec 2.0 Base [5]	95.04M	LS 960 hr	75.18	6.02	6.08	5.74	6.43	46.95	96.23	0.0233	14.81	92.35	88.30	24.77	63.43	2.55	93.9	9.77	7.50	10.5	98.00	69.6
HuBERT Base [6]	94.68M	LS 960 hr	81.42	5.11	5.88	5.41	6.42	46.69	96.30	0.0736	15.53	98.34	88.53	25.20	64.92	2.58	93.9	9.36	7.47	8.0	98.50	70.9
WavLM Base	94.70M	LS 960 hr	84.51	4.69	4.55	4.84	6.21	42.81	96.79	0.0870	20.74	98.63	89.38	22.86	65.94	2.58	94.0	10.37	7.42	8	98.00	72.0
- w/o denoising task	94.70M	LS 960 hr	84.39	4.91	6.03	4.85	6.08	43.61	96.79	0.0799	21.03	98.42	88.69	23.43	65.55	2.56	93.9	9.91	7.43	**7.5**	97.75	71.7
- w/o structure modification	94.68M	LS 960 hr	84.74	4.61	4.72	5.22	6.80	42.88	96.79	0.0956	20.03	98.31	88.56	24.00	65.60	2.58	94.0	10.29	7.45	8.4	99.00	71.9
WavLM Base+	94.70M	Mix 94k hr	89.42	4.07	3.50	3.92	5.59	38.32	97.37	**0.0988**	24.25	99.00	90.58	21.20	68.65	2.63	94.3	10.85	7.40	8.1	99.00	73.4
wav2vec 2.0 Large [5]	317.38M	LL 60k hr	86.14	5.65	5.62	4.75	3.75	44.69	96.66	0.0489	12.48	95.28	87.11	27.31	65.64	2.52	94.0	10.02	7.63	15.8	97.25	70.4
HuBERT Large [6]	316.61M	LL 60k hr	90.33	5.98	5.75	3.53	3.62	44.08	95.29	0.0353	20.01	98.76	89.81	21.76	67.62	2.64	94.2	10.45	7.22	9.0	99.25	72.2
WavLM Large	316.62M	Mix 94k hr	**95.49**	**3.77**	**3.24**	**3.06**	**3.44**	**32.27**	**97.86**	0.0886	**26.57**	**99.31**	**92.21**	**18.36**	**70.62**	**2.70**	**94.5**	**11.19**	7.30	9.0	99.00	**74.6**

3. wav2vec 2.0, Hubert and WavLM are three most effective models.

S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin et al., "Superb: Speech processing universal performance benchmark," arXiv preprint arXiv:2105.01051, 2021.

S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao et al., "Wavlm: Large-scale self-supervised pre- training for full stack speech processing," IEEE Journal of Selected Topics in Signal Processing, vol. 16, no. 6, pp. 1505–1518, 2022.

Results

Comparation within SSL models

Method	#Params	Corpus	Speaker			Content					Semantics				ParaL	Generation						Overall
			SID	ASV	SD	PR	ASR	OOD-ASR	KS	QbE	ST	IC	SF		ER	SE		SS	VC			
			Acc ↑	EER ↓	DER ↓	PER ↓	WER ↓	WER ↓	Acc ↑	MTWV ↑	BLEU ↑	Acc ↑	F1 ↑	CER ↓	Acc ↑	PESQ ↑	STOI ↑	SI-SDRi ↑	MCD ↓	WER ↓	ASV ↑	Score ↑
FBANK	0	-	8.5E-4	9.56	10.05	82.01	23.18	63.58	8.63	0.0058	2.32	9.10	69.64	52.94	35.39	2.55	93.6	9.23	8.47	38.3	77.25	43.2
PASE+ [44]	7.83M	LS 50 hr	37.99	11.61	8.68	58.87	25.11	61.56	82.54	0.0072	3.16	29.82	62.14	60.17	57.86	2.56	93.9	9.87	8.66	30.6	63.20	51.5
APC [30]	4.11M	LS 360 hr	60.42	8.56	10.53	41.98	21.28	63.12	91.01	0.0310	5.95	74.69	70.46	50.89	59.33	2.56	93.4	8.92	8.05	27.2	87.25	59.2
VQ-APC [29]	4.63M	LS 360 hr	60.15	8.72	10.45	41.08	21.20	63.56	91.11	0.0251	4.23	74.48	68.53	52.91	59.66	2.56	93.4	8.44	7.84	22.4	94.25	59.5
NPC [33]	19.38M	LS 360 hr	55.92	9.40	9.34	43.81	20.20	61.66	88.96	0.0246	4.32	69.44	72.79	48.44	59.08	2.52	93.1	8.04	7.86	30.4	94.75	59.0
Mockingjay [35]	85.12M	LS 360 hr	32.29	11.66	10.54	70.19	22.82	65.27	83.67	6.6E-04	4.45	34.33	61.59	58.89	50.28	2.53	93.4	9.29	8.29	35.1	79.75	51.0
TERA [34]	21.33M	LS 960 hr	57.57	15.89	9.96	49.17	18.17	58.49	89.48	0.0013	5.66	58.42	67.50	54.17	56.27	2.54	93.6	10.19	8.21	25.1	83.75	57.2
DeCoAR 2.0 [37]	89.84M	LS 960 hr	74.42	7.16	6.59	14.93	13.02	53.62	94.48	0.0406	9.94	90.80	83.28	34.73	62.47	2.47	93.2	8.54	7.83	17.1	90.75	66.3
modified CPC [53]	1.84M	LL 60k hr	39.63	12.86	10.38	42.54	20.18	62.54	91.88	0.0326	4.82	64.09	71.19	49.91	60.96	2.57	93.7	10.40	8.41	26.2	71.00	56.9
wav2vec [39]	32.54M	LS 960 hr	56.56	7.99	9.9	31.58	15.86	55.86	95.59	0.0485	6.61	84.92	76.37	43.71	59.79	2.53	93.8	9.30	7.45	10.1	98.25	63.5
vq-wav2vec [40]	34.15M	LS 960 hr	38.80	10.38	9.93	33.48	17.71	60.66	93.38	0.0410	5.66	85.68	77.68	41.54	58.24	2.48	93.6	8.16	**7.08**	13.4	**100.00**	61.8
wav2vec 2.0 Base [5]	95.04M	LS 960 hr	75.18	6.02	6.08	5.74	6.43	46.95	96.23	0.0233	14.81	92.35	88.30	24.77	63.43	2.55	93.9	9.77	7.50	10.5	98.00	69.6
HuBERT Base [6]	94.68M	LS 960 hr	81.42	5.11	5.88	5.41	6.42	46.69	96.30	0.0736	15.53	98.34	88.53	25.20	64.92	2.58	93.9	9.36	7.47	8.0	98.50	70.9
WavLM Base	94.70M	LS 960 hr	84.51	4.69	4.55	4.84	6.21	42.81	96.79	0.0870	20.74	98.63	89.38	22.86	65.94	2.58	94.0	10.37	7.42	8	98.00	72.0
- w/o denoising task	94.70M	LS 960 hr	84.39	4.91	6.03	4.85	6.08	43.61	96.79	0.0799	21.03	98.42	88.69	23.43	65.55	2.56	93.9	9.91	7.43	**7.5**	97.75	71.7
- w/o structure modification	94.68M	LS 960 hr	84.74	4.61	4.72	5.22	6.80	42.88	96.79	0.0956	20.03	98.31	88.56	24.00	65.60	2.58	94.0	10.29	7.45	8.4	99.00	71.9
WavLM Base+	94.70M	Mix 94k hr	89.42	4.07	3.50	3.92	5.59	38.32	97.37	**0.0988**	24.25	99.00	90.58	21.20	68.65	2.63	94.3	10.85	7.40	8.1	99.00	73.4
wav2vec 2.0 Large [5]	317.38M	LL 60k hr	86.14	5.65	5.62	4.75	3.75	44.69	96.66	0.0489	12.48	95.28	87.11	27.31	65.64	2.52	94.0	10.02	7.63	15.8	97.25	70.4
HuBERT Large [6]	316.61M	LL 60k hr	90.33	5.98	5.75	3.53	3.62	44.08	95.29	0.0353	20.01	98.76	89.81	21.76	67.62	2.64	94.2	10.45	7.22	9.0	99.25	72.2
WavLM Large	316.62M	Mix 94k hr	**95.49**	**3.77**	**3.24**	**3.06**	**3.44**	**32.27**	**97.86**	0.0886	**26.57**	**99.31**	**92.21**	**18.36**	**70.62**	**2.70**	**94.5**	**11.19**	7.30	9.0	99.00	**74.6**

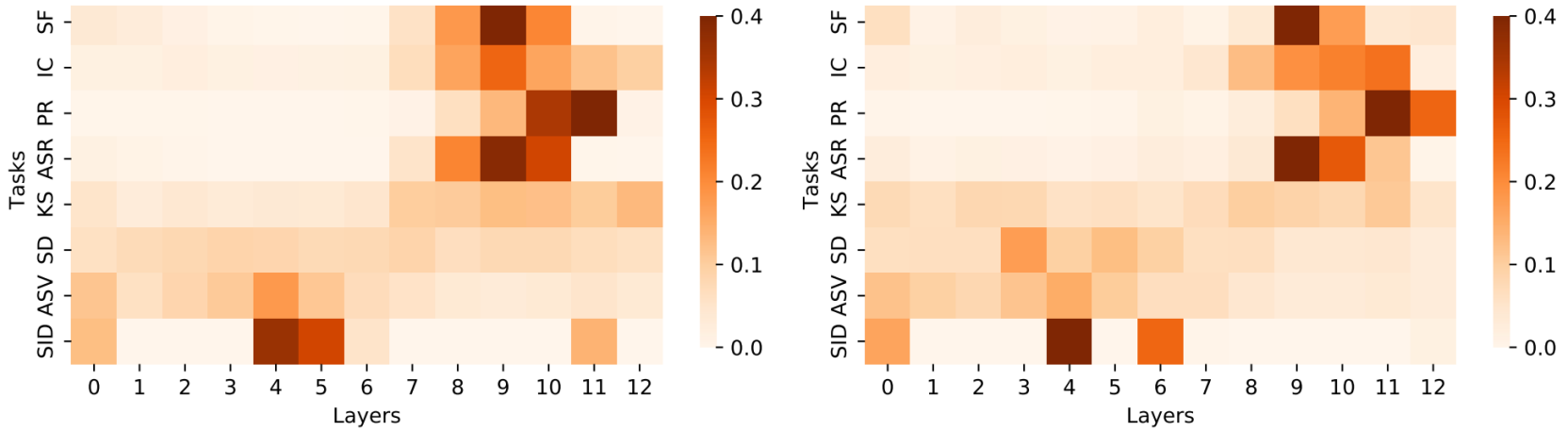
4. Scaling law: Larger model and larger dataset produce better performance.

S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin et al., "Superb: Speech processing universal performance benchmark," arXiv preprint arXiv:2105.01051, 2021.

S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao et al., "Wavlm: Large-scale self-supervised pre- training for full stack speech processing," IEEE Journal of Selecte Topics in Signal Processing, vol. 16, no. 6, pp. 1505–1518, 2022.

Results

Weight Analysis of each layer in SSL models



- Shallow layers tend to contain more speaker-related information
- While Deep layers contain more contextual or semantic related information

S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao et al., "Wavlm: Large-scale self-supervised pre- training for full stack speech processing," IEEE Journal of Selecte Topics in Signal Processing, vol. 16, no. 6, pp. 1505–1518, 2022.

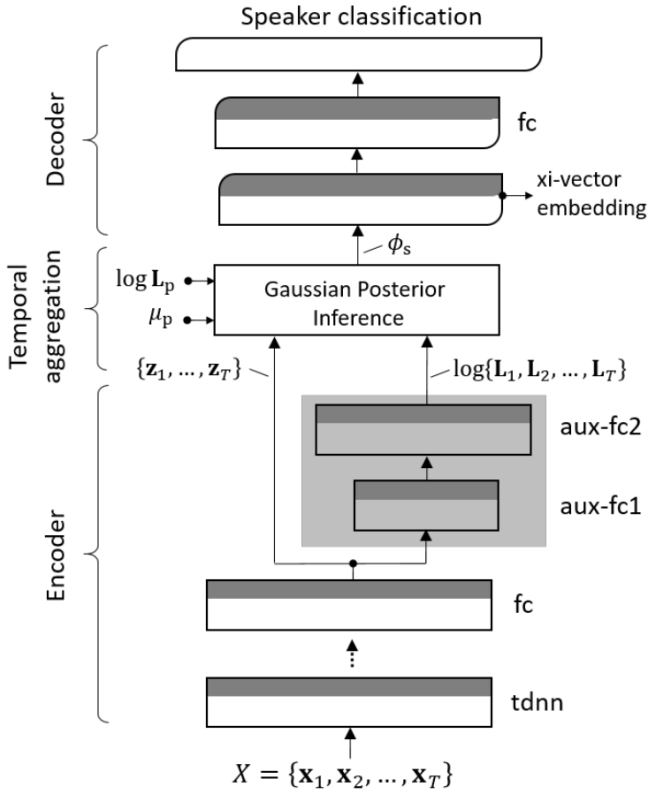
Summarization

From this study, we learn:

- What is Self-Supervised Learning models.
- How to use representation features from SSL models
- Comparison with different SSL models.
- SSL model is able to disentangle information from raw inputs.

Inspirations

How to apply to SSL models for modeling uncertainty information



Xi-vector

1. We could replace tdnn with SSL models to get better feature representation

2. Using the classification information of Hubert as pseduo-classification mean target to compute uncertainty.

Lee, Kong Aik, Qiongqiong Wang, and Takafumi Koshinaka. "Xi-vector embedding for speaker recognition." *IEEE Signal Processing Letters* 28 (2021): 1385-1389.