



Boulder

Speech & Music; Modeling



[YouTube Playlist](#)

Maziar Raissi

Assistant Professor

Department of Applied Mathematics

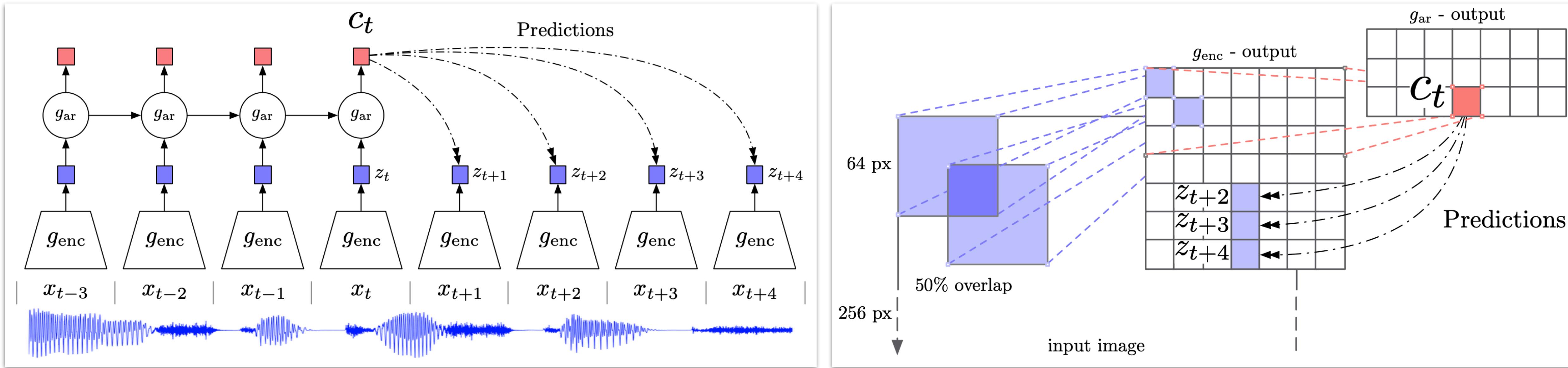
University of Colorado Boulder

maziar.raissi@colorado.edu



Representation Learning with Contrastive Predictive Coding

Boulder



$g_{\text{enc}} \rightarrow$ non-linear encoder (e.g., strided convolutional layers with resnet blocks)

$x_t \rightarrow$ input observation

$z_t = g_{\text{enc}}(x_t) \rightarrow$ latent representation

$g_{\text{ar}} \rightarrow$ autoregressive model (e.g., GRUs)

$c_t = g_{\text{ar}}(z_{\leq t}) \rightarrow$ context latent representation (summarizing all $z_{\leq t}$ in the latent space)

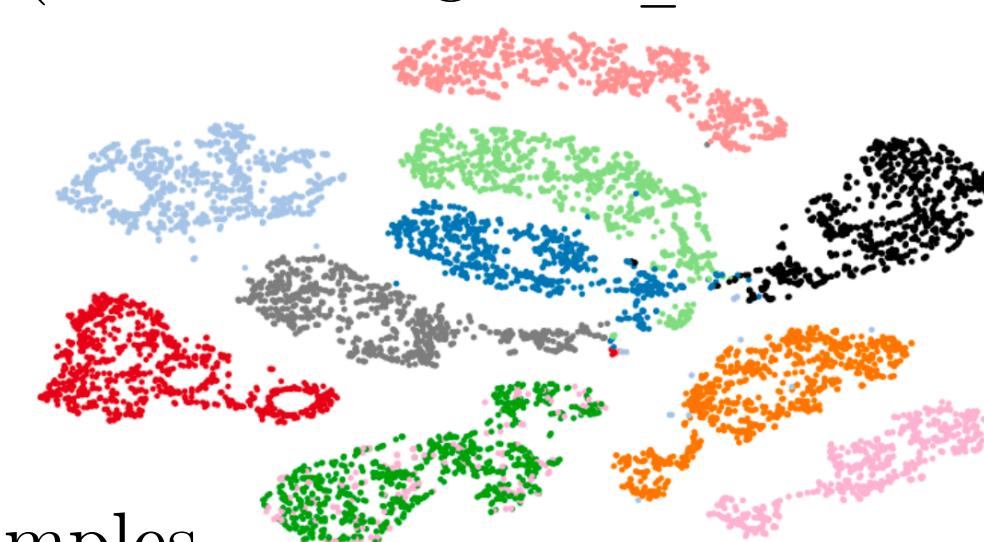
$$f_k(x_{t+k}, c_t) := \exp(\underbrace{z_{t+k}^T W_k}_{\hat{z}_{t+k}} c_t)$$

InfoNCE Loss

$$\mathcal{L}_N = -\mathbb{E}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right]$$

$X = \{x_1, x_2, \dots, x_N\} \rightarrow$ set of N random samples

Containing one positive sample from $p(x_{t+k}|c_t)$ and $N - 1$ negative samples from the proposal distribution $p(x_{t+k})$



→

t-SNE visualization of speech (each color represents a different speaker)

Speech, images, text and reinforcement learning!

Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding." *arXiv preprint arXiv:1807.03748* (2018).

Mutual Information Estimation

The optimal value for $f(x_{t+k}, c_t)$ is proportional to $\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}$!

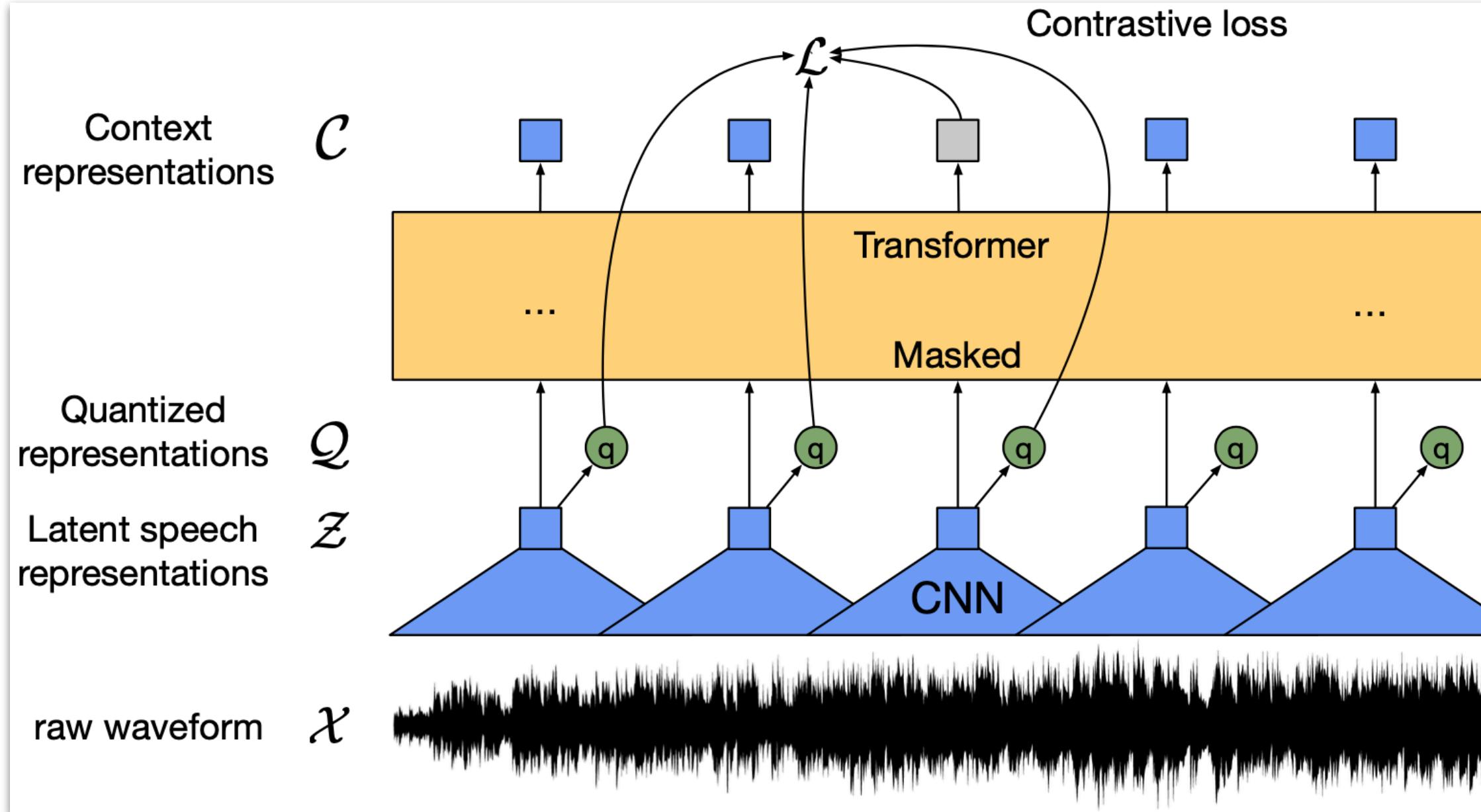
No need to predict future observations x_{t+k} directly with a generative model $p(x_{t+k}|c_t)$. InfoNCE (Noise Contrastive Estimation) relieves the model from modeling the high dimensional distributions x_{t+k} .

$$I(x_{t+k}; c_t) \geq \log N - \mathcal{L}_N$$

$$I(x; c) = \sum_{x,c} p(x, c) \log \frac{p(x|c)}{p(x)} \rightarrow \text{mutual information}$$

Minimizing \mathcal{L}_N implies maximizing $I(x_{t+k}; c_t)$.

wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations


[YouTube Video](#)


$$f : \mathcal{X} \mapsto \mathcal{Z}$$

└ multi-layer convolutional feature encoder

$\mathcal{X} \rightarrow$ input raw audio

$\mathcal{Z} = (z_1, z_2, \dots, z_T) \rightarrow$ latent speech representations

$$g : \mathcal{Z} \mapsto \mathcal{C}$$

└ transformer

$\mathcal{C} = (c_1, c_2, \dots, c_T) \rightarrow$ representations capturing information from the entire sequence

Instead of fixed positional embeddings which encode absolute positional information, use a convolutional layer which acts as relative positional embedding.

$$\mathcal{Z} \xrightarrow{Q} \mathcal{Q}$$

quantization module
 $\mathcal{Q} = (q_1, q_2, \dots, q_T)$

diversity loss: encourage the model to use the codebook entries equally often

Quantization Module

$G \rightarrow$ number of codebooks/groups

$V \rightarrow$ number of entries

$$e \in \mathbb{R}^{V \times d/G}$$

	avg. WER	std.
Continuous inputs, quantized targets (Baseline)	7.97	0.02
Quantized inputs, quantized targets	12.18	0.41
Quantized inputs, continuous targets	11.18	0.16
Continuous inputs, continuous targets	8.58	0.08

Choose one entry/row from each codebook e and concatenate the resulting vectors e_1, \dots, e_G and apply a linear transformation $\mathbb{R}^d \rightarrow \mathbb{R}^f$ to obtain $q \in \mathbb{R}^f$. The Gumbel softmax enables choosing discrete codebook entries in a fully differentiable way!

$$z \mapsto l$$

$z \rightarrow$ feature encoder output

$$l \in \mathbb{R}^{G \times V} \rightarrow \text{logits}$$

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

probability of choosing the v -th codebook entry for group g

$\tau \rightarrow$ non-negative temperature

$$n = -\log(-\log(u)) \rightarrow \text{Gumbel noise} \quad u \sim U(0, 1)$$

Forward pass: $i = \arg \max_j p_{g,j} \rightarrow$ codeword i

Backward pass: true gradient of the Gumbel softmax outputs

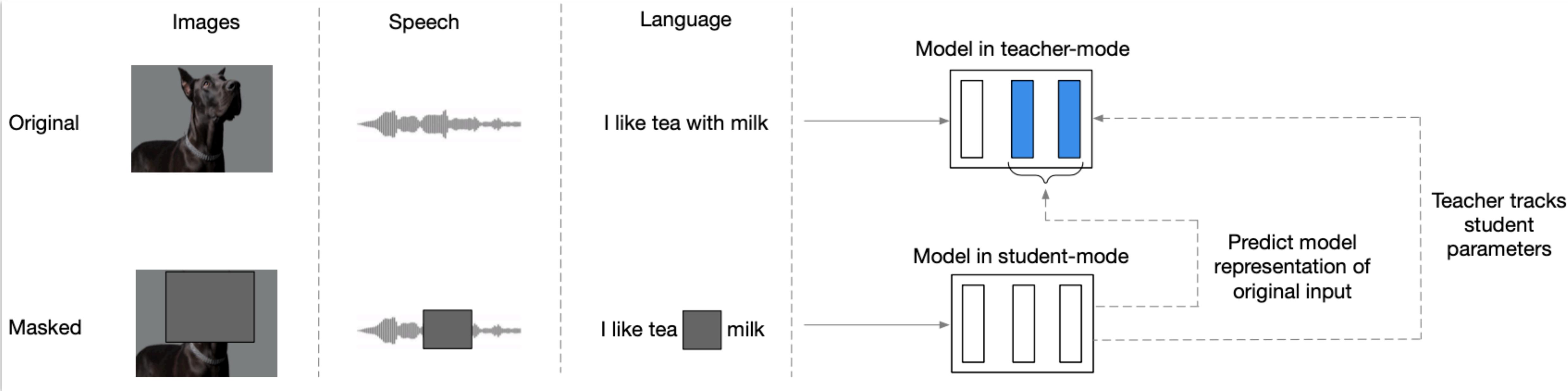
$$\mathcal{L}_m = -\log \frac{\exp(\text{sim}(\mathbf{c}_t, \mathbf{q}_t)/\kappa)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp(\text{sim}(\mathbf{c}_t, \tilde{\mathbf{q}})/\kappa)} \quad \text{sim}(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T \mathbf{b} / \|\mathbf{a}\| \|\mathbf{b}\|$$

contrastive loss: identify the true quantized latent speech representation for a masked time step within a set of distractors.

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



data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language



Modality-specific encoding of the input data

Computer Vision: sequence of patches (each spanning 16×16 pixels)

Speech: encoded using a multi-layer 1D CNN mapping 16kHz waveform to 50Hz representations

Text: byte-pair encoded sub-word units (learned embedding vectors)

Teacher Parametrization

exponentially moving average (EMA)

$$\Delta \leftarrow \tau \Delta + (1 - \tau) \theta$$

Targets

$$y_t = \frac{1}{K} \sum_{l=L-K+1}^L \hat{a}_t^l$$

$\hat{a}_t^l \rightarrow$ normalized output of block l at time-step t

Averaging top K blocks

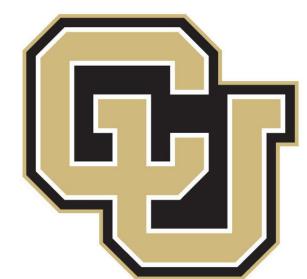
$$\mathcal{L}(y_t, f_t(x)) = \begin{cases} \frac{1}{2}(y_t - f_t(x))^2 / \beta & |y_t - f_t(x)| \leq \beta \\ (|y_t - f_t(x)| - \frac{1}{2}\beta) & \text{otherwise} \end{cases}$$

Smooth L1 loss

	ViT-B	ViT-L
MoCo v3 (Chen et al., 2021b)	83.2	84.1
DINO (Caron et al., 2021)	82.8	-
BEiT (Bao et al., 2021)	83.2	85.2
MAE (He et al., 2021)	83.6	85.9
SimMIM (Xie et al., 2021)	83.8	-
MaskFeat (Wei et al., 2021)	84.0	85.7
data2vec	84.2	86.2

	MNLI	QNLI	RTE	MRPC	QQP	STS-B	CoLA	SST	Avg.
<i>Base models</i>									
BERT (Devlin et al., 2019)	84.0/84.4	89.0	61.0	86.3	89.1	89.5	57.3	93.0	80.7
Baseline (Liu et al., 2019)	84.1/83.9	90.4	69.3	89.0	89.3	88.9	56.8	92.3	82.5
data2vec	83.2/83.0	90.9	67.0	90.2	89.1	87.2	62.2	91.8	82.7
+ wav2vec 2.0 masking	82.8/83.4	91.1	69.9	90.0	89.0	87.7	60.3	92.4	82.9

	Unlabeled data	LM	Amount of labeled data				
			10m	1h	10h	100h	960h
<i>Base models</i>							
wav2vec 2.0 (Baevski et al., 2020b)	LS-960	4-gram	15.6	11.3	9.5	8.0	6.1
HuBERT (Hsu et al., 2021)	LS-960	4-gram	15.3	11.3	9.4	8.1	-
WavLM (Chen et al., 2021a)	LS-960	4-gram	-	10.8	9.2	7.7	-
data2vec	LS-960	4-gram	12.3	9.1	8.1	6.8	5.5



Boulder



Questions?

[YouTube Playlist](#)
