CONTRASTIVE PREDICTIVE CODING SUPPORTED FACTORIZED VARIATIONAL AUTOENCODER FOR UNSUPERVISED LEARNING OF DISENTANGLED SPEECH REPRESENTATIONS

Janek Ebbers[*], *Michael Kuhlmann*[*], *Tobias Cord-Landwehr, Reinhold Haeb-Umbach*

Paderborn University, Department of Communications Engineering, Paderborn, Germany
{ebbers,kuhlmann,cord,haeb}@nt.upb.de

ABSTRACT

In this work we address disentanglement of style and content in speech signals. We propose a fully convolutional variational autoencoder employing two encoders: a content encoder and a style encoder. To foster disentanglement, we propose adversarial contrastive predictive coding. This new disentanglement method does neither need parallel data nor any supervision. We show that the proposed technique is capable of separating speaker and content traits into the two different representations and show competitive speaker-content disentanglement performance compared to other unsupervised approaches. We further demonstrate an increased robustness of the content representation against a train-test mismatch compared to spectral features, when used for phone recognition.

Index Terms— speech disentanglement, unsupervised learning, contrastive learning, autoencoder

1. INTRODUCTION

Disentangling factors of variation in data recently attracted increased interest for many modalities. Disentangled representations are considered useful in two ways. First, they can improve performance for various downstream tasks, which are learned on a small amount of labeled data. In particular, it can yield improved robustness against train-test mismatches if the factors that are informative about the task can be successfully disentangled from the variations caused by a domain shift. Second, in a disentangled representation certain factors can be modified while keeping the rest fixed, e.g., changing the lighting in an image without changing the content. Learning disentangled representations with no or only little supervision is particularly interesting, because it can make use of the vast amounts of unannotated data available in the world.

In this paper we address disentanglement of speech signals by separating long-term variations, referred to as style, and short-term variations, referred to as content. The proposed factorized variational autoencoder (FVAE) employs two encoders to extract two disjoint representations, namely, a style embedding and a sequence of content embeddings, which are jointly decoded to reconstruct the input signal. To restrict the content embeddings to only capture short-term variations, we propose an adversarial regularization based on contrastive predictive coding (CPC) [1], which is completely unsupervised. The basic idea is to penalize mutual information between a current and a future content embedding. Therefore, slowly changing variations need to be captured by the style embedding.

While the proposed approach is rather general, we here focus on its capability to disentangle speaker and linguistic attributes. Our proposed model is trained on speech data requiring neither phonetic labels nor speaker labels. We also do not require parallel data, where the same linguistic content is spoken by different speakers. Compared to other unsupervised approaches, we demonstrate superior disentanglement performance in terms of downstream phone classification and speaker verification tasks. Particularly, we show that the learned content embeddings are largely speaker invariant and achieve an increased robustness against a train-test mismatch compared to spectral features. While we focus on disentanglement in this paper, we also provide listening examples showing that the model can be used to perform voice conversion (VC) when decoding content embeddings with an exchanged style embedding.

The rest of the paper is structured as follows. After we discussed related work in Sec. 2, an overview of CPC is given in Sec. 3. Then, Sec. 4 presents our proposed FVAE and how we use CPC to support disentanglement. Finally, experiments are shown in Sec. 5 and conclusions are drawn in Sec. 6.

2. RELATED WORK

There are many works focusing on unsupervised disentanglement of all latent factors of a generative model [2, 3, 4]. Those works are mainly applied to toy-like image data sets, e.g., 2D shapes [5], where the generating factors are well defined. Other works tackle supervised disentanglement of a single factor using an adversarial classifier in the latent space [6, 7].

While the above works target other modalities, there are several recent works tackling disentangled speech representation learning from non-parallel data. Many works, e.g., [8, 9, 10, 11], focus on extracting a speaker independent content representation, while representing the speaker identity as a one-hot encoding. Others also use speaker specific decoders [12, 13]. Therefore, these works can neither be used for style extraction nor for voice conversion with unknown target speakers. Also, speaker labels are required.

Unsupervised approaches to speaker-content disentanglement are proposed in [14, 15, 16, 17]. None of these works use an explicit disentanglement objective as proposed in this paper. The authors of [15] propose to encourage disentanglement by using instance normalization (IN) in the content encoder, which removes, to some extent, static signal properties such as speaker attributes. The AutoVC model [17] relies on a carefully tuned bottleneck such that ideally all content information can be stored in the content embedding but none of the speaker-related information. In [18], the AutoVC model was extended to an unsupervised disentanglement of timbre, pitch, rhythm and content. The VQVC [16] achieves disentanglment by applying a bottleneck in terms of vector quantization (VQ). The factorized hierarchical variational autoencoder (FHVAE) proposed in [14] unsupervisedly disentangles "sequence-level" ($>200\,\mathrm{ms}$) and "segment-level" ($<200\,\mathrm{ms}$) attributes, by restricting sequence-level embeddings to be rather static within an utterance while also putting a bottleneck on the segment-level embeddings.

[*]First two authors contributed equally. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 282835863. Computational resources were provided by the Paderborn Center for Parallel Computing.

3. CONTRASTIVE PREDICTIVE CODING

Given a sequence $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_T]$ of feature vectors $\mathbf{y}_t$, which are in our case features of a speech signal, CPC [1] aims at extracting the mutual information from two segments $\mathbf{Y}_t{=}[\mathbf{y}_{t-c}, \dots, \mathbf{y}_{t+c}]$ and $\mathbf{Y}_{t-\tau}$, where c denotes a one-sided context length and τ a shift between the segments. For this purpose, the segments are encoded into the embeddings $\mathbf{h}_t{=}f_{\text{cpc}}(\mathbf{Y}_t)$ and $\mathbf{h}_{t-\tau}{=}f_{\text{cpc}}(\mathbf{Y}_{t-\tau})$ such that $\mathbf{h}_{t-\tau}$ allows the prediction of the future embedding $\mathbf{h}_t$:

$$\hat{\mathbf{h}}_t = g_\tau(\mathbf{h}_{t-\tau})$$

with $g_\tau(\cdot)$ denoting the projection head that predicts τ steps ahead. The CPC model is trained using a contrastive loss [1]:

$$L_{\text{cpc}} = -\frac{1}{T-\tau}\sum_{t=\tau+1}^{T}\frac{\exp(\mathbf{h}_t^{\mathrm{T}}\hat{\mathbf{h}}_t)}{\sum\limits_{\widetilde{\mathbf{h}}_t\in\mathcal{H}_t}\exp(\widetilde{\mathbf{h}}_t^{\mathrm{T}}\hat{\mathbf{h}}_t)}\ , \tag{1}$$

where $\mathcal{H}_t$ is the set of candidate embeddings $\{\mathbf{h}_t^{(b)} \mid 1 \leq b \leq B\}$ in the mini-batch of size B. Note that Eq. (1) equals a cross entropy loss including a softmax where the logits are given as the inner product of the predicted embedding $\hat{\mathbf{h}}_t$ and the candidate embeddings $\widetilde{\mathbf{h}}_t \in \mathcal{H}_t$. Hence, the model is essentially trained to be able to correctly classify the true future embedding out of a couple of candidates. The number of steps τ that the model predicts into the future controls the kind of mutual information that is encoded. If τ is small, i.e., the segments are very close to each other, the model probably learns to recognize content attributes, e.g., whether they are parts of the same acoustic unit. However, if τ is large, i.e., the segments are further apart, the mutual information the model has to recognize are primarily the static properties such as speaker attributes.

4. CONTRASTIVE PREDICTIVE CODING SUPPORTED FACTORIZED VARIATIONAL AUTOENCODER

To learn disentangled representations of style and content, we propose a fully convolutional FVAE which is illustrated in Fig. 1. The FVAE employs two encoders: a content encoder to encode content information from an input sequence $\mathbf{X}{=}[\mathbf{x}_1, \dots, \mathbf{x}_T]$ into a sequence of content embeddings $\mathbf{Z}{=}[\mathbf{z}_1, \dots, \mathbf{z}_N]$, and a style encoder to extract style traits into a style embedding $\bar{\mathbf{s}}$.

As input signal representation $\mathbf{X}$ we extract $F{=}80$ log mel-filter bank energy (F-Bank) features for each frame of a short-time Fourier transform (STFT) using an audio sample rate of 16 kHz, a frame length of 50 ms and a hop-size of 12.5 ms. Each log-mel band is normalized by subtracting the global mean and dividing by the global standard deviation, which are determined on a training set.

The encoders and the decoder are one-dimensional convolutional neural networks (CNNs) as shown in Fig. 2. The style encoder uses $K_{\text{o}}{=}S_{\text{o}}{=}1$ yielding frame level outputs $\mathbf{S}{=}[\mathbf{s}_1, \dots, \mathbf{s}_T]$ on which a global average pooling (GAP) over time is applied to obtain an utterance-level style embedding $\bar{\mathbf{s}}$. The content encoder uses $K_{\text{o}}{=}S_{\text{o}}{=}S_{\text{ds}}$ with downsampling being performed when $S_{\text{ds}}{>}1$ such that $N{=}\lceil T/S_{\text{ds}}\rceil$. The concatenated embeddings $\begin{bmatrix}\bar{\mathbf{s}} & \dots & \bar{\mathbf{s}}\\ \mathbf{z}_1 & \dots & \mathbf{z}_N\end{bmatrix}$ are forwarded to the decoder network which outputs an input signal reconstruction $\hat{\mathbf{X}}$. The input layer of the decoder upsamples the embeddings of length N back to length T by using $K_{\text{i}}{=}S_{\text{i}}{=}S_{\text{ds}}$. The training objective for reconstruction is given by the mean squared error (MSE):

$$L_{\text{rec}} = \frac{1}{T}||\hat{\mathbf{X}} - \mathbf{X}||_2^2\ .$$

To support style extraction, we apply an auxiliary CPC loss $L_{\text{cpc}}^{(\mathbf{S})}$ to the style encoder output, which is the loss from Eq. (1) with variables $\mathbf{h}_t$ replaced by $\mathbf{s}_t$ here. As we aim to extract utterance-level style information, we choose a large $\tau{=}80$ which corresponds to a shift of 1 s. Furthermore, the projection head $g_\tau(\cdot)$ is chosen to be the identity $\hat{\mathbf{s}}_t{=}\mathbf{s}_{t-\tau}$ such that the style encoder is encouraged to extract similar embeddings within the same utterance and orthogonal embeddings in different utterances.

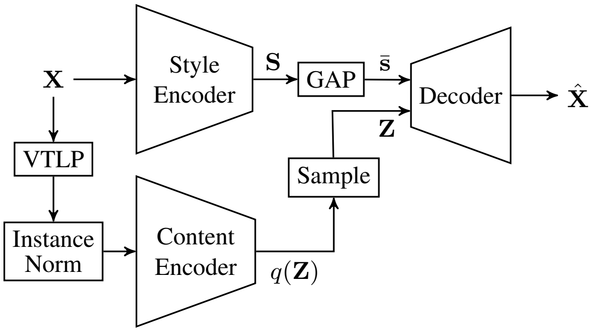

Fig. 1: Factorized VAE.

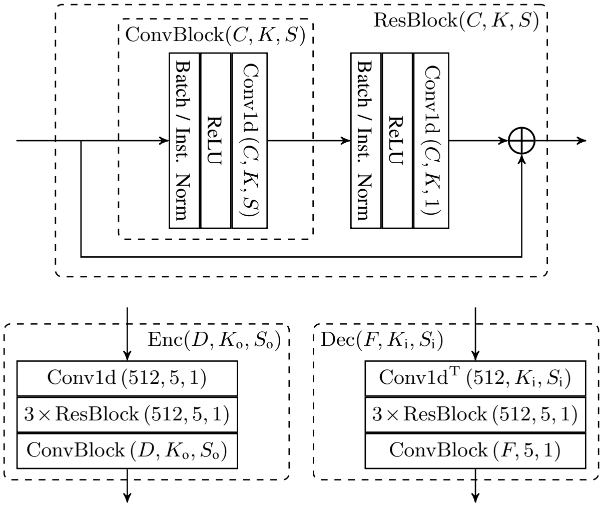

Fig. 2: CNN Architectures: $\text{Conv1d}\,(C, K, S)$ denotes a one-dimensional convolutional layer with C output channels, kernel size of K and striding of S. $\text{Conv1d}^{\mathrm{T}}$ denotes a one-dimensional transposed convolutional layer.

Naturally, the proposed model would tend to access all the required signal information through $\mathbf{Z}$ while ignoring $\mathbf{s}$ as there is usually much more capacity in a sequence of embeddings $\mathbf{Z}$ than in a single embedding $\bar{\mathbf{s}}$. Therefore, the challenge is to prevent the model from also encoding style properties of the signal into $\mathbf{Z}$ but make the model access it through $\bar{\mathbf{s}}$.

VAEs [19] have been used to put an information bottleneck on the content embedding which has shown to improve disentanglement [8]. Here, $\mathbf{z}_n$ is interpreted as a stochastic variable with prior $p(\mathbf{z}_n){=}\mathcal{N}(\mathbf{z}_n; \mathbf{0}, \mathbf{I})$ and an approximate posterior $q(\mathbf{z}_n){=}\mathcal{N}(\mathbf{z}_n; \boldsymbol{\mu}_n, \text{diag}(\boldsymbol{\sigma}_n^2))$, with the content encoder providing $\boldsymbol{\mu}_n$ and $\log\left(\boldsymbol{\sigma}_n^2\right)$. The content embeddings that are forwarded into the decoder are sampled as $\mathbf{z}_n{\sim}q(\mathbf{z}_n)$ using the reparameterization trick [19] during training, while being set to $\mathbf{z}_n{=}\boldsymbol{\mu}_n$ in test mode. The Kullback-Leibler divergence (KLD)

$$L_{\text{kld}} = \frac{1}{N}\sum_{n=1}^{N}\text{KL}(q(\mathbf{z}_n)||p(\mathbf{z}_n))$$

that is added to the VAE objective prefers the posterior $q(\mathbf{z}_n)$ to be uninformative which helps encoding information into $\bar{\mathbf{s}}$ rather than $\mathbf{Z}$. However, it also harms reconstruction which is why it should not be weighted too high either.

Another simple additional measure to encourage disentanglement is to remove or distort utterance-level properties in the input of the content encoder. Then, $\mathbf{X}$ cannot be reconstructed without accessing information via the style encoder. First, during training we distort speaker properties using vocal tract length perturbation (VTLP) [20]. VTLP was proposed to increase speaker variability when training speech recognition systems. For this purpose, the center bins of the mel-filter-banks are randomly remapped using a piecewise linear warping function. Second, we perform IN [21] on the content input, i.e., the individual log-mel bands of each input signal are normalized across time to zero mean and unit variance. We perform IN also in the hidden layers of the content encoder, instead of batch normalization used in the other sub-networks, which has been found useful to encourage speaker-content disentanglement [15].

However, none of the above measures explicitly prevents the content encoder from encoding style. To enforce disentanglement, the authors of [11] suggested to employ a jointly trained adversarial speaker classifier on content embeddings to make the content encoder drop information revealing the speaker identity. However, the adversarial speaker classifier has two main disadvantages. First, although it does not require text annotations, it still requires speaker annotations. Second, it does not scale to large unbalanced databases with a huge number of speakers as the classification task itself becomes too uncertain to obtain useful adversarial gradients.

Therefore, in this work we propose adversarial CPC as an alternative which is fully unsupervised and independent of the (unobserved) number of speakers. Hence, this approach has the potential to scale to large unlabeled databases.

We train a CPC encoder $\mathbf{H}{=}[\mathbf{h}_1,\ldots,\mathbf{h}_T]{=}f_{\text{acpc}}(\mathbf{Y}_\mathbf{Z})$ on the content encoder's output $\mathbf{Y}_\mathbf{Z}{=}[\mathbf{y}_{\mathbf{z}_1},\ldots,\mathbf{y}_{\mathbf{z}_N}]{=}\begin{bmatrix}\mathbf{M}_\mathbf{Z}\\ \log\mathbf{V}_\mathbf{Z}\end{bmatrix}$ with $\mathbf{M}_\mathbf{Z}{=}[\boldsymbol{\mu}_{\mathbf{z}_1},\ldots,\boldsymbol{\mu}_{\mathbf{z}_N}]$ being the sequence of posterior means and $\mathbf{V}_\mathbf{Z}{=}[\boldsymbol{\sigma}^2_{\mathbf{z}_1},\ldots,\boldsymbol{\sigma}^2_{\mathbf{z}_N}]$ being the sequence of posterior variances. For the CPC encoder we use the architecture $f_{\text{cpc}}{=}\text{Dec}(D_\mathbf{h}, K_{\text{ds}}, S_{\text{ds}})$ similar to the VAE decoder with $D_\mathbf{h}{=}128$. As before, we are interested in extracting global style information which is why we again choose $\tau{=}80$ corresponding to 1 s and an identity projection $\hat{\mathbf{h}}_t{=}\mathbf{h}_{t-\tau}$. The CPC encoder is trained to minimize the CPC loss (1), which is referred to as $L^{(\mathbf{Z})}_{\text{cpc}}$ here. By adding the negative of the CPC loss to the FVAE objective, the content encoder tries to maximize it, i.e., the FVAE content encoder and the CPC encoder operate adversarially here. This prevents the FVAE content encoder from encoding mutual information into the content embeddings which are 1 s apart (or further) eventually preventing it from encoding speaker attributes and other static style properties.

Finally, our complete training objective is given as

$$L_{\text{fvae}} = L_{\text{rec}} + \beta L_{\text{kld}} + \lambda_\mathbf{S} L^{(\mathbf{S})}_{\text{cpc}} - \lambda_\mathbf{Z} L^{(\mathbf{Z})}_{\text{cpc}} \ .$$

5. EXPERIMENTS

Experiments are performed on the Librispeech corpus [22] which is derived from audiobooks and contains, for each speaker, utterances from one or multiple chapters. Here, *train-clean-100* and *train-clean-360* subsets are used for training. These sets contain 100 h and 360 h of clean speech from 251 and 921 speakers, respectively. From *train-clean-100*, however, we only use 60 % of the speakers' utterances for training and leave 2×20 % for validation and evaluation of a closed-speaker scenario. In the following we therefore refer to the train, validation and test portions as *train-clean-60*, *dev-closed* and *test-closed*, respectively. Validation and evaluation of an open-speaker scenario is done with the official *dev-clean* and *test-clean* sets, respectively. In all trainings, utterances shorter than 2 s are discarded. Remaining utterances are cut into equally long segments such that no segment is longer than 4 s.

We compare our proposed method with the following disentanglement baselines that are also scalable and can be trained fully unsupervised, i.e., also without any supervision on speaker labels.

Adaptive instance normalization (AdaIN): The authors of [15] proposed to use IN in the content encoder to normalize style information. The decoder uses AdaIN [23] which applies IN and then adaptively computes shifts and scales of an affine transformation from a style embedding extracted by a style encoder.

Dimensionality bottleneck: We already motivated that it is crucial to put an information bottleneck on the content embeddings to achieve disentanglement. The authors of [17] found that disentanglement can be achieved by only carefully tuning the dimensionality of the latent content representation. We therefore compare our proposed CPC supported FVAE with both the AutoVC model proposed in [17] as well as with a simple dimensionality tuning in our proposed FVAE without applying any supporting CPC losses. AutoVC usually uses a pre-trained speaker encoder trained using speaker labels. To make the model fully unsupervised for a fair comparison, we here replace it with a vanilla CPC encoder $\mathbf{H} = f_{\text{cpc}}(\mathbf{X})$ pre-trained on F-bank features $\mathbf{X}$ with $f_{\text{cpc}} = \text{Enc}(128, 1, 1)$, $\tau{=}80$ and $g_\tau(\mathbf{h}) = \mathbf{h}$ to extract style embeddings.

FVAE and baseline models are trained on the combined *train-clean-60* and *train-clean-360* sets for 10^5 update steps using mini-batches of 32 segments. When using an adversarial CPC model, it is exclusively updated for three additional steps after each joint update of FVAE and CPC. Further, an initial warm up of 400 exclusive FVAE steps followed by 1200 exclusive adversarial CPC steps is performed. Adam [24] is used for optimization with a learning rate of $5{\cdot}10^{-4}$ and gradient clipping is applied using thresholds of 10, 20 and 2 for VAE encoders, decoder and CPC encoder, respectively. After training, the checkpoint which achieves lowest reconstruction error on the validation set is used to report results on the test set. For the AdaIN and AutoVC baselines we use the model implementations released by the authors. Content and speaker embedding sizes of $D_\mathbf{z}{=}32$ and $D_\mathbf{s}{=}128$ are used for all models.

The vanilla CPC model is, similarly to the disentanglement models, trained on the combined *train-clean-60* and *train-clean-360* sets for $5 \cdot 10^4$ update steps using mini-batches of 64, a learning rate of 10^{-3} and gradient clipping at a threshold of 2.

Disentanglement performance is evaluated as follows.

Post-hoc classifiers: After training, downstream phone and speaker classifiers are trained on the models' content embeddings $\mathbf{Z}$. As the content embeddings are supposed to be informative about the linguistic content, we aim to obtain a low phone error rate (PER). In contrast, they are also expected to be speaker invariant which should therefore not allow speaker classification. Hence, a high speaker error rate (SER) indicates good disentanglement. Similar evaluations were made, e.g., in [15, 17]. The architecture of the classifiers is similar to the decoder with $f^{(\mathbf{Z})}_{\text{phn}}{=}\text{Dec}(40, S_{\text{ds}}, S_{\text{ds}})$ and $f^{(\mathbf{Z})}_{\text{spk}}{=}\text{Dec}(251, S_{\text{ds}}, S_{\text{ds}})$, where 40 is the number of target phones and 251 the number of target speakers. If not stated otherwise, the classifiers are trained on the *train-clean-60* set with the phone classifier being validated and evaluated on *dev-clean* and *test-clean* and the speaker classifier on *dev-closed* and *test-closed*, respectively. Phone alignments for training are imported from Kaldi [25]. Training is performed for $5 \cdot 10^4$ update steps using mini-batches of 64, a

Table 1: Disentanglement performance measured by downstream PER on the content embedding (lower is better), SER on the content embedding (higher is better) and speaker verification on the style embedding (lower is better). All values are stated in %. Best result in a block is shown in bold.

#	Model	IN	VTLP	S_{ds}	$\lambda_{\mathbf{Z}}$	$\lambda_{\mathbf{S}}$	PER($\mathbf{Z}$)	SER($\mathbf{Z}$)	EER($\bar{\mathbf{s}}$)	
									WC	AC
1	F-Bank	✗	✗	-	-	-	17.7	1.7	17.3	22.4
2	F-Bank	✓	✓	-	-	-	15.3	-	-	-
3	I-Vector	-	-	-	-	-	-	-	4.1	6.7
4	CPC	✗	✗	-	-	-	-	-	2.4	6.7
5	AdaIN [15]	✓	✗	8	-	-	**21.4**	37.4	7.8	11.2
6	AutoVC [17]	✗	✗	16	-	-	22.0	**52.9**	**2.4***	**6.7***
7	FVAE	✓	✓	8	0	0	**16.6**	18.8	10.1	13.9
8	FVAE	✓	✓	16	0	0	17.8	38.0	9.2	12.9
9	FVAE	✓	✓	32	0	0	19.8	65.6	8.8	12.2
10	FVAE	✓	✓	64	0	0	28.3	**86.2**	**6.6**	**10.0**
11	FVAE	✓	✓	8	1	0	17.4	**51.3**	7.9	11.8
12	FVAE	✓	✓	8	0	1	**16.7**	17.3	2.2	**6.6**
13	FVAE	✓	✓	8	1	1	17.4	48.1	2.1	6.7
14	FVAE	✓	✗	8	1	1	17.8	44.0	2.2	6.7
15	FVAE	✗	✗	8	1	1	20.0	28.6	**2.0**	6.7

*Scores from pre-trained CPC which is used by AutoVC as style encoder.

learning rate of 10^{-3} and gradient clipping at a threshold of 20. The checkpoint which achieves highest accuracy on the validation set is used to report results on the test set. As reference, we also report performance of classifiers trained with F-bank features.

Speaker Verification: Speaker verification aims to recognize a pair of utterances as either from the same speaker or from different speakers by evaluating the similarity of style embeddings. Here, the distance between two embeddings is measured by their cosine distance where a pair is classified as same speaker when the distance falls below a certain threshold. Performance is reported as the equal error rate (EER) which is the error rate for the threshold where the false acceptance rate and false rejection rate are equal. We aim at a low EER as we expect to have similar style embeddings for utterances from the same speaker and different ones for utterances from different speakers. As Librispeech is not a standard set for speaker verification, we randomly generate a verification task. For each utterance from *test-clean* we sample an utterance from every other speaker to generate a negative pair yielding a total of $\sim$100 k negative trials. For positive pairs we consider two setups: In the *within chapter (WC)* setup all possible combinations of a speakers' utterances within the same chapter are considered totaling $\sim$50 k positive pairs. In the *across chapter (AC)* setup we randomly replace one of the utterances in each WC pair, if possible[1], by an utterance from another chapter. As reference we report the performance of i-vectors [26] which have been trained on *train-clean-60* using Kaldi. Note that i-vector training requires speaker supervision though and can hence be considered a topline. We further report performance of the vanilla CPC model which is unsupervised.

Table 1 shows results for different models and different values for the hyper parameters S_{ds} (subsampling factor determining the dimensionality bottlenck), $\lambda_{\mathbf{Z}}$ (adversarial CPC weight), $\lambda_{\mathbf{S}}$ (auxiliary CPC weight), usage of IN and usage of VTLP. All FVAEs have been trained using β=0.01. The first and second block show reference and baseline systems, respectively. The third block presents a dimensionality tuning in our proposed architecture without CPC support which can also be viewed a baseline for the proposed additional CPC losses. The last block reports results when using CPC supported training.

Table 2: Mismatched PER when only trained on male speakers. Best result in column is shown bold.

Features	w/o VTLP			w/ VTLP		
	male	female	all	male	female	all
F-Bank	**24.5**	42.8	33.5	**23.4**	30.2	26.7
FVAE	24.8	**31.14**	**28.2**	24.5	**28.4**	**26.4**

Evaluation of the content embeddings shows that all models are able to significantly improve speaker invariance compared to F-bank features. Model 10 gives the best SER result achieving 86.2%. However, this comes with the cost of a high PER. Comparing PERs, it can be seen that in most configurations the proposed FVAE significantly outperforms the baselines. In block three it can be seen that a smaller bottleneck due to a higher subsampling factor leads to an increase in both PER and SER as expected. Here, model 8 gives a good balance between PER, which is still not far off, and speaker invariance, which is decently high. Compared to model 8, the models 11 and 13 from block four use a smaller S_{ds} but use adversarial CPC. It can be seen that this yields an improvement of both PER and speaker invariance. Comparing the last three models 13, 14 and 15 shows that PER and speaker invariance both highly benefit from IN and slightly benefit from VTLP.

Moving on to the evaluation of the style embeddings, it can be seen that WC verification generally achieves a significantly higher performance than AC verification. This can be explained due to differences in recording conditions between utterances from different chapters, such as volume and reverberation. Note that unsupervisedly trained style encoders are meant to capture such style differences and, hence, a worse verification performance in the AC setup is not necessarily a sign of bad quality here. It can be seen that in both setups the best verification performance can be achieved by our proposed FVAE when using an auxiliary CPC loss on the style encoders' output (models 12-15) allowing to even outperform i-vectors and vanilla CPC style embeddings.

Next, we evaluate the robustness of the proposed content feature extraction in the case of little training data suffering from a train-test mismatch. Such a scenario is simulated by training a phone classifier on only a subset of 20 male speakers from *train-clean-60* and test on female speakers. Results are reported in Tab. 2 for both, when using VTLP and when not using VTLP in classifier training as well as in FVAE training (models 13,14 in Tab.1). It can be seen that our proposed model outperforms F-Bank features in mismatched evaluations. As expected the performance difference is, however, much smaller when VTLP is used. However, note that the male-female mismatch is only a proof of concept. Other mismatches might not be compensated by a simple data augmentation such as VTLP.

Finally, we provide listening examples[2] showing that the model is able to perform zero-shot VC. An extensive investigation and evaluation of the model's VC capabilities is, however, left to the future.

6. CONCLUSIONS

The proposed CPC support for training an FVAE conducts disentanglement of speaker and content-induced variations. Its training is fully unsupervised and does not even require knowledge of speaker labels. Compared to other unsupervised disentanglement approaches, superior disentanglement can be achieved in terms of down stream phone recognition and speaker verification. Further, the proposed content embedding extraction allows to obtain an increased robustness against a train-test mismatch when used for phone classification.

[1] Some speakers only have utterances from a single chapter.

[2] `go.upb.de/acpcvc`

7. REFERENCES

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

[2] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner, "beta-VAE: Learning basic visual concepts with a constrained variational framework.," *International Conference on Learning Representations*, vol. 2, no. 5, pp. 6, 2017.

[3] Hyunjik Kim and Andriy Mnih, "Disentangling by factorising," in *International Conference on Machine Learning*, 2018, pp. 2649–2658.

[4] Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud, "Isolating sources of disentanglement in variational autoencoders," in *Advances in Neural Information Processing Systems*, 2018, pp. 2610–2620.

[5] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner, "dSprites: Disentanglement testing sprites dataset," *URL https://github. com/deepmind/dsprites-dataset*, 2017.

[6] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato, "Fader networks: Manipulating images by sliding attributes," in *Advances in Neural Information Processing Systems*, 2017, pp. 5967–5976.

[7] Naama Hadad, Lior Wolf, and Moni Shahar, "A two-step disentanglement method," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 772–780.

[8] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2016, pp. 1–6.

[9] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu, "Neural discrete representation learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 6306–6315.

[10] Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, and Nobukatsu Hojo, "ACVAE-VC: Non-parallel voice conversion with auxiliary classifier variational autoencoder," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 9, pp. 1432–1443, 2019.

[11] Ju-Chieh Chou, Cheng-Chieh Yeh, Hung-Yi Lee, and Lin-shan Lee, "Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations," *Proceedings of Interspeech 2018*, pp. 501–505, 2018.

[12] Adam Polyak and Lior Wolf, "Attention-based wavenet autoencoder for universal voice conversion," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2019*. IEEE, 2019, pp. 6800–6804.

[13] Keonnyeong Lee, In-Chul Yoo, and Dongsuk Yook, "Voice conversion using cycle-consistent variational autoencoder with multiple decoders," *arXiv preprint arXiv:1909.06805*, 2019.

[14] Wei-Ning Hsu, Yu Zhang, and James Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," in *Advances in neural information processing systems*, 2017, pp. 1878–1889.

[15] Ju-Chieh Chou and Hung-Yi Lee, "One-shot voice conversion by separating speaker and content representations with instance normalization," *Proceedings of Interspeech 2019*, pp. 664–668, 2019.

[16] D. Wu and H. Lee, "One-shot voice conversion by vector quantization," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2020*, 2020, pp. 7734–7738.

[17] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson, "AutoVC: Zero-shot voice style transfer with only autoencoder loss," in *International Conference on Machine Learning*, 2019, pp. 5210–5219.

[18] Kaizhi Qian, Yang Zhang, Shiyu Chang, David Cox, and Mark Hasegawa-Johnson, "Unsupervised speech decomposition via triple information bottleneck," *arXiv preprint arXiv:2004.11284*, 2020.

[19] Diederik P. Kingma and Max Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[20] Navdeep Jaitly and Geoffrey E Hinton, "Vocal tract length perturbation (vtlp) improves speech recognition," in *Proceedings of ICML Workshop on Deep Learning for Audio, Speech and Language*, 2013, vol. 117.

[21] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[22] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "LibriSpeech: an asr corpus based on public domain audio books," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2015*. IEEE, 2015, pp. 5206–5210.

[23] Xun Huang and Serge Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.

[24] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[25] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The Kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, 2011, number CONF.

[26] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.