Pattern Recognition Letters 181 (2024) 64–69

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Deep neural networks for automatic speaker recognition do not learn supra-segmental temporal features

Daniel Neururer [a,1], Volker Dellwo [b], Thilo Stadelmann [a,c,*,1]

[a] Centre for Artificial Intelligence, Zurich University of Applied Sciences, Technikumstrasse 71, 8400, Winterthur, Switzerland
[b] Department of Computational Linguistics, University of Zurich, Andreasstrasse 15, 8050, Zurich, Switzerland
[c] European Centre for Living Technology (ECLT), Ca' Bottacin, Dorsoduro 3911, 30123, Venice, Italy

ARTICLE INFO

Editor: Li Liu

Keywords:
Speaker verification
Speaker clustering
Dynamic features
Prosodic features
Deep learning
Explainable AI (XAI)

ABSTRACT

While deep neural networks have shown impressive results in automatic speaker recognition and related tasks, it is dissatisfactory how little is understood about what exactly is responsible for these results. Part of the success has been attributed in prior work to their capability to model supra-segmental temporal information (SST), i.e., learn rhythmic-prosodic characteristics of speech in addition to spectral features. In this paper, we (i) present and apply a novel test to quantify to what extent the performance of state-of-the-art neural networks for speaker recognition can be explained by modeling SST; and (ii) present several means to force respective nets to focus more on SST and evaluate their merits. We find that a variety of CNN- and RNN-based neural network architectures for speaker recognition do not model SST to any sufficient degree, even when forced. The results provide a highly relevant basis for impactful future research into better exploitation of the full speech signal and give insights into the inner workings of such networks, enhancing explainability of deep learning for speech technologies.

1. Introduction

Deep neural networks (DNNs) have become extremely effective in speaker recognition (SR) and its sub-tasks like speaker verification (SV), identification (SI) or clustering (SC) [1]. Despite this success, deep learning remains driven by empiricism [2], and the available theoretical insights into its workings [3] all the more underline what is yet not understood about how and why DNNs arrive at such a high performance [4], leaving much room for improved explainability of such models [5] also to guide future research.

Meanwhile, the key to human top performance in SR (specifically, in challenging environments) is to make use of a comprehensive variety of spectro-temporal acoustic-phonetic information in speech [6,7]. Particularly, short-term spectral information, equating to frame-based acoustic information (**FBA**) in automatic systems, is supplemented in humans by supra-segmental temporal information (**SST**), also referred to as speech prosody. SST varies between individuals [8,9] and is beneficial for automatic SR systems [10]. Latter authors provided first evidence that succeeding in adequately modeling SST in addition to FBA holds the potential for an order of magnitude less SR errors. Further evidence has been provided using convolutional and recurrent deep learning architectures [11–13], claiming that the achieved gains are due to the superior sequence modeling capabilities of the DNNs that are "successfully capturing prosodic information" [13] — without explicitly testing this. In a similar vein [14,15]: Zhao et al. [14] argue that their BLSTM-enhanced [16] SV DNN is "*supposedly* phonetically aware" and modeling "context information, which *could* facilitate the ResNet to [. . .] suppress the environmental variations", *because* BLSTM layers have the capability to model long ranges. Yet, these claims have never been verified.

If DNNs would not model SST adequately (but achieve their superior results otherwise by focusing on FBA alone), this would imply that the predicted performance gains [10] are still to be realized. More specifically, if it could be quantified to what extent state-of-the-art DNN-based SR systems actually do or do not exploit SST, this would (a) add explanation to a high-performing but opaque class of SR models, (b) show specific directions for targeted future research (concretely targeting the modeling of SST, if it turns out to be under-exploited), and (c) verify theoretical as well as empirical findings in earlier studies, practically guiding future developments. This way, the found discrepancies between DNNs theoretical capabilities and their practical workings are not just uncovered, but can be reconciled in the future.

In this paper, we experimentally analyze this hypothesis of superior modeling of supra-segmental temporal features through contemporary

[*] Corresponding author at: Centre for Artificial Intelligence, Zurich University of Applied Sciences, Technikumstrasse 71, 8400, Winterthur, Switzerland.
E-mail addresses: neud@zhaw.ch (D. Neururer), volker.dellwo@uzh.ch (V. Dellwo), stdm@zhaw.ch (T. Stadelmann).

[1] Contributed equally to this work.

https://doi.org/10.1016/j.patrec.2024.03.016
Received 12 April 2023; Received in revised form 24 July 2023; Accepted 21 March 2024
Available online 26 March 2024

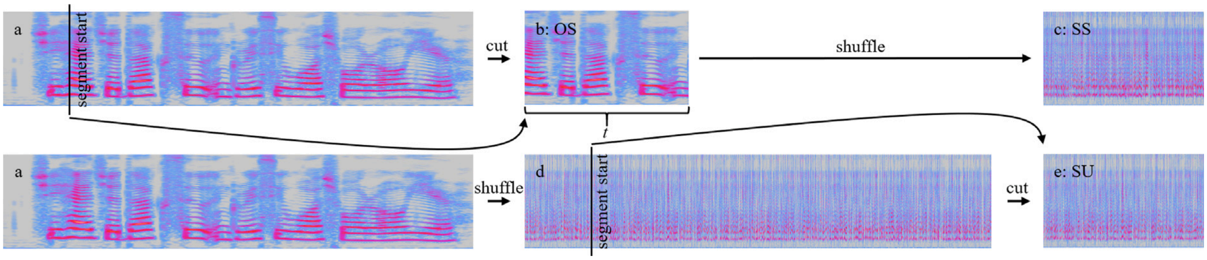

Fig. 1. Segment creation with and without SST: Starting (a) from a spectrogram per varying-length *utterance*, we extract *segments* of fixed length t from a starting point according to 3 segment-drawing strategies as follows. Original Segment (OS): just cut out (b) the respective part; Shuffled within Segment (SS): additionally, shuffle (c) the columns of the previous output; Shuffled within Utterance (SU): globally shuffle all columns (d) prior to cutting (e).

DNNs made in prior work, suspecting that it happens less than assumed. Our motivation is to better understand and improve DNNs in their ability to model fundamental aspects of the speech signal inherently. Hence, we present a novel approach to quantify what amount of sequence modeling is actually occurring, and find the hypothesis of superior SST modeling has to be rejected: Current DNNs lazily rest on phone-level acoustic features alone (cf. Section 2). We offer further analyses of this undesirable *"deep cheating"* phenomenon (cf. Section 3) by gradually removing the speaker-discriminant information contained in FBA features through innovative experimental setups, thus forcing models to rely on other sources of information. Still they do not switch to model available SST, as evidenced in extensive experiments using diverse state-of-the-art DNN architectures for SV using TIMIT [17] and VoxCeleb [18] benchmarks.

To the best of our knowledge, this paper represents the first systematic study into the modeling of SST by DNNs. Its contributions stand further out in several ways: First, our results explain what DNNs for SR do and do not model, namely that they overfit on the easily exploitable FBA to the point where they nearly ignore SST information, rectifying earlier published conjectures. Second, this opens strategic directions for future SR research that are perpendicular to other current research trends, namely to inquire into better exploitation of SST (cf. Section 4) to realize the predicted performance gain of one order of magnitude less errors [10]. Third, the developed benchmark suite of experimental protocol, test metric and established results makes progress in this direction quantifiable.

2. Time scrambling approach to quantify SST exploitation

2.1. Objective and related work

We study SR DNNs that receive spectrogram-like input and are interested in quantifying to which extent they rely on FBA, i.e., spectral characteristics as contained in a single frame of MFCCs [19], and to which extent they exploit SST, i.e., information that is contained in the trajectory over many frames, like intonation and rhythm. Works like [13,14] assert that such sequence learning is happening automatically because it is a reasonable explanation for the achieved superior results, given the general sequence learning capabilities of the models. On the other hand, Soleymani et al. [20] do not rely on CNNs to pick up SST automatically just via filters that extend in time: In addition to feeding mel-spectrograms into convolutional layers to extract FBA, they feed hand-crafted prosodic features for late fusion into the DNN to achieve prosody-enhanced SV.

The survey by Bai and Zhang [21] shows that these two views are omnipresent in the literature: Some works use DNN architectures inspired by computer vision [22,23], thereby asserting that exploiting dynamics will happen via filters and average pooling along the time axis in the same way that such architectures pick up image information along the x-axis automatically. According to Bai and Zhang, this is "the most common [temporal] pooling function". At the same time, they survey works that deal explicitly with integrating FBA over time, either via collecting statistics [24,25], applying self-attention [26] or performing a trainable soft clustering of FBA [27]. In this section, we lay the foundation for an informed decision on these options by quantifying how well DNNs *inherently* capture SST in spectrogram-like input without additional explicit care for temporal dynamics.

2.2. Methodology and experimental setup

The authors of [10] used a simple test to probe human reliance on SST: They randomly shuffled the columns of a spectrogram, re-synthesis the result back to audio, and conducted human SR experiments. This way, they left all FBA information intact but removed all SST within said spectrograms. Inspired by this time scrambling approach, we aim to quantify the amount of actual SST exploitation in a DNN by comparing SR performance on original input data with performance on time-scrambled input that contains no SST: Starting from a mel-spectrogram input (front-end processing similar to [13] for comparability), this is achieved by randomizing the order of speech frames, i.e., columns in the spectrogram, thereby removing all original sequence information. The anticipated drop in recognition performance would confirm an anti-proportionally good exploitation of SST through DNNs, while no drop would at least mean that the DNN can compensate missing SST equivalently with the remaining FBA information, hence does not rely on SST exploitation.

Specifically, we consider three ways to extract fixed-length time-scrambled *segments* from a variable-length *utterance* (cf. Fig. 1) according to the following reasoning: **OS** segments will contain FBA and SST, offering a DNN all options to learn these lower- and higher-level features. **SS** segments will, due to a random trajectory of frames, contain no speaker-specific dynamics, offering DNNs only the option to learn about FBA. **SU** segments constitute certain middle ground, biased towards FBA: While still containing no SST, they draw frames from the full utterance rather than a limited segment, offering a richer sample of FBA.

We carried out the following initial experiments on the TIMIT database which provides a controlled acoustic recording environment and normalized utterances (630 speakers under clean studio conditions, 10 sentences per speaker of on average 3s length, 462 speakers in the training set). This laboratory setup enabled us to first study voice recognition performance of a DNN model *per se*, without complications induced through the environment (background noise, cross talk, etc.). As our main interest is in understanding and improving DNN modeling for SR (in contrast to improving a complete SR pipeline), we used three different DNN architectures to sample the space of simple to advanced models without explicit care for temporal modeling: (a) As a simple baseline, an adapted version of the vanilla convolutional network (CNN) of [12], sped up by a CosFace loss function [28] instead of the computationally heavy KL divergence; (b) as the first model that claimed to model prosody, the recurrent neural network (RNN) of [13]; (c) as a recent architecture, the ResNet34s (ResNet) of [27] that innately contains a GhostVLAD layer to aggregate FBA per segment, adapted to the experimental setup of the previous models with respect

Table 1
SC results on TIMIT [MR μ/σ]. Bold font indicates best results per model, cell coloring scales with quality per model. Cells marked by ⇓ are discussed in the text.

↓ training / test →		OS	SU	SS
CNN [12]	OS	a⇓ **0.00 σ0.00**	9.75 σ0.94	a,f⇓ 9.00 σ2.15
	SU	8.50 σ2.42	e⇓ 0.50 σ0.61	1.75 σ0.61
	SS	a⇓ 9.00 σ1.66	1.00 σ0.50	b⇓ 1.25 σ0.00
RNN [13]	OS	d⇓ 1.25 σ1.12	2.75 σ0.94	f⇓ 2.75 σ0.50
	SU	3.75 σ1.37	**0.00 σ0.00**	2.50 σ1.58
	SS	2.00 σ1.00	1.25 σ0.79	d⇓ 0.25 σ0.50
ResNet [27]	OS	d⇓ **1.00 σ0.94**	8.25 σ4.78	f⇓ 11.50 σ4.29
	SU	2.50 σ1.77	**1.00 σ0.50**	3.00 σ1.27
	SS	2.75 σ0.94	1.25 σ1.12	d⇓ **1.00 σ0.94**
F-ResNet [29]	OS	11.50 σ2.15	37.50 σ4.18	33.75 σ4.18
	SU	16.50 σ2.42	5.75 σ1.70	4.25 σ1.00
	SS	15.50 σ2.57	6.75 σ1.50	j⇓ **3.75 σ0.79**

Table 2
SV results on TIMIT [EER μ/σ]. As with the SC results on TIMIT in Table 1, the F-ResNet is out of competition here due to different front-end processing.

↓ training / test →		OS	SU	SS
CNN [12]	OS	c⇓ 6.38 σ0.12	12.02 σ0.51	f⇓ 11.90 σ0.46
	SU	8.55 σ0.49	g-i⇓ 5.55 σ0.06	6.12 σ0.12
	SS	8.16 σ0.42	**5.33 σ0.18**	c⇓ 5.78 σ0.16
RNN [13]	OS	d⇓ **3.53 σ0.07**	4.19 σ0.09	f⇓ 3.90 σ0.12
	SU	3.99 σ0.16	3.78 σ0.10	3.66 σ0.13
	SS	4.00 σ0.07	3.89 σ0.06	d,g-i⇓ 3.54 σ0.05
ResNet [27]	OS	**4.96 σ0.19**	10.34 σ1.56	f⇓ 9.21 σ1.15
	SU	6.59 σ0.25	6.25 σ0.23	6.37 σ0.35
	SS	5.89 σ0.25	6.11 σ0.31	g-i⇓ 5.80 σ0.11
F-ResNet [29]	OS	12.20 σ0.25	23.41 σ1.73	20.47 σ1.50
	SU	15.12 σ0.84	10.46 σ0.28	9.69 σ0.20
	SS	15.91 σ0.90	9.86 σ0.11	j⇓ **8.95 σ0.13**

to front-end processing (see below); (d) additionally, to account for recent developments in state-of-the-art approaches, the Fast ResNet-34 (F-ResNet) of [29][2] that also serves as a competitive baseline in the VoxSRC SR benchmarking efforts [1], used here with its original parameters also for front-end processing (and hence not directly comparable to the first three models). Utmost care has been taken to keep all not explicitly mentioned parameters (here or below) for, e.g., front-end processing, DNN architecture and training, embedding extraction, and classification equal to the respective original work in order to allow for comparisons. Hence, the first three models allow a comparison with prior work on modeling dynamic voice features with DNNs, while the fourth model allows a comparison with the current state-of-the art. We expected all models to suffer performance loss in increasing magnitude when fed with shuffled frames within segments due to increasing capability to model SST.

For training of the CNN, RNN, and ResNet model, to warrant backward comparability with prior work, in each of 128 epochs we draw one $t = 1$ s long segment (to account for short utterances [30]) from a random location per utterance in the training set, using a batch size of 100 and otherwise a similar experimental setup as [13]. For the F-ResNet, to keep forward comparability with current SR benchmarking efforts, we keep all hyperparameters similar to the experimental setup in [29], specifically using $t = 2$ s long training segments ($t = 4$ s for evaluation), 500 epochs and a batch size of 800. We repeat this for each of the OS/SS/SU strategies to draw segments with/without SST, thus training three separate versions of each model. We use these models to extract embeddings for two downstream tasks: For SV, we pair every sentence in the TIMIT test set with every other sentence therein, and evaluate the equal error rate (EER) as the standard metric used for SV [1], resulting in 2.82 mio. pairwise comparisons. For SC, we perform hierarchical clustering of 2 utterances (comprising 2 and 8 concatenated sentences) per 40 speakers as in [13], measured by the misclassification rate (MR) as the established measure [11] (6400 pairwise comparisons). For each reported EER/MR, we average over 5 train/test runs and report mean and SD.

2.3. Results and discussion

Results are shown in Tables 1 and 2. For the analysis, we first focus on SC results (Table 1) and original timing vs. segment-wise shuffled (rows/columns marked with OS/SS), ignoring the rest. With reference to the lowercase letters in the cells marked by ⇓, we can say: (a) Looking at the CNN model only, everything appears as expected as best results ($MR = 0$ in all 5 runs, the best result ever reported on this benchmark) are achieved for training and testing with OS, and worst results are achieved when SS is used for training *or* evaluation. (b) But already looking at training *and* testing using SS raises doubts w.r.t. actual SST exploitation: $MR = 1.25$ is a very good result for the task, still outperforming the previous state of the art [13], but not using any SST (as it has been removed from the data). (c) Taking, secondly, SV results (Table 2) into account, these doubts are confirmed. For the CNN model, training/testing using SS/SS outperforms OS/OS by 0.6 difference in EER. (d) A similar picture is seen for the RNN on both tasks and the ResNet on SC, where SS/SS either outperforms or (within σ) equals OS/OS. (e) When adding SU for training and/or testing into the picture, this tendency is confirmed: SU/SU outperforms SS/SS and is almost on par with OS/OS (cf. a). (f) However, models do pick up something about inter-frame relationships as in cross conditions like OS training and SS testing, a performance drop is evident for the CNN and ResNet (not so for the RNN). We argue that this is rather the effect of mismatched train/test conditions as is usual in any machine learned model; previous discussion (specifically, that best results are achieved

[2] Sourced from https://github.com/clovaai/voxceleb_trainer.

Table 3
SV results on VoxCeleb (left) and noise-vocoded (middle) as well as resynthesized (right) TIMIT [EER μ/σ of 5 runs].

		VoxCeleb			TIMIT-NV			TIMIT-Syn		
↓ training / test →		OS	SU	SS	OS	SU	SS	OS	SU	SS
CNN [12]	OS	g⇓ **25.75 σ0.13**	37.23 σ0.74	36.96 σ0.78	32.56 σ0.62	35.32 σ0.46	35.41 σ0.55	46.24 σ0.18	48.94 σ0.15	48.97 σ0.23
	SU	32.70 σ0.34	27.04 σ0.34	27.99 σ0.30	35.16 σ0.52	h⇓ **30.39 σ0.30**	30.91 σ0.47	47.26 σ0.15	45.98 σ0.34	46.16 σ0.27
	SS	33.26 σ0.29	27.91 σ0.32	28.50 σ0.28	35.25 σ0.69	30.63 σ0.38	31.23 σ0.27	47.14 σ0.22	45.88 σ0.12	i⇓ **45.66 σ0.12**
RNN [13]	OS	g⇓ **20.67 σ0.23**	30.67 σ0.36	30.00 σ0.32	h⇓ **19.34 σ0.16**	27.20 σ0.42	26.12 σ0.44	i⇓ **40.39 σ0.07**	44.29 σ0.65	42.43 σ1.40
	SU	26.20 σ0.18	22.02 σ0.10	23.57 σ0.09	22.95 σ0.24	21.48 σ0.40	21.15 σ0.25	43.63 σ0.35	41.93 σ0.26	41.64 σ0.25
	SS	28.28 σ1.30	26.30 σ0.59	26.58 σ0.84	22.82 σ0.40	21.89 σ0.25	21.04 σ0.12	43.62 σ0.21	42.55 σ0.34	41.53 σ0.23
ResNet [27]	OS	g⇓ **12.49 σ0.15**	34.11 σ0.54	32.19 σ0.39	h⇓ **21.12 σ0.43**	37.83 σ1.17	36.57 σ1.45	i⇓ **40.33 σ1.32**	47.28 σ2.06	46.60 σ2.02
	SU	22.05 σ0.43	19.08 σ0.26	20.02 σ0.16	27.03 σ0.63	23.38 σ0.41	24.02 σ0.25	43.44 σ0.86	42.97 σ0.51	42.65 σ0.59
	SS	20.74 σ0.46	21.02 σ0.34	20.36 σ0.23	27.25 σ1.37	23.57 σ0.46	23.32 σ0.58	42.48 σ0.45	43.07 σ0.72	41.59 σ0.36
F-ResNet [29]	OS	g⇓ **2.39 σ0.05**	25.02 σ1.21	23.45 σ1.29	k⇓ 24.72 σ0.54	37.17 σ0.57	34.54 σ0.66	k⇓ **39.06 σ0.50**	47.52 σ0.58	46.48 σ1.02
	SU	11.00 σ0.57	6.69 σ0.11	6.58 σ0.15	31.46 σ0.86	22.40 σ0.42	21.67 σ0.49	42.85 σ0.47	40.10 σ0.11	40.19 σ0.10
	SS	10.85 σ0.43	7.02 σ0.24	6.60 σ0.19	31.44 σ1.11	22.04 σ0.12	k⇓ **21.24 σ0.17**	43.23 σ0.29	40.44 σ0.18	40.36 σ0.20

using SU in several cases) has shown that (almost) nothing useful is extracted from the trajectory. (j) The F-ResNet, which uses the front-end processing and architectural parameters optimized for SV on the by orders of magnitude larger VoxCeleb dataset, has expected difficulties with the tiny TIMIT benchmark (which could be cured by proper pre-training, but would not help the purpose of these experiments). It hence ran out of competition here, but will play a major role in later experiments (see Section 3). For now, it suffices to say that it behaved quite similar to the ResNet. That the best scores are achieved using the SS/SS setup on both tasks can be explained with the general lack of sufficient amounts of training data, specifically when using longer segments, and that in this regime SS provides most FBA information per segment (i.e., the higher sample efficiency can partially make up for lacking training data).

Summarizing, we found that completely randomizing the order of speech frames in segments used for training and evaluation still produced state-of-the-art or even better SV (e.g., SU/SU for the RNN) and SC (e.g., SS/SS for the RNN) results. This is notwithstanding the obviously also state-of-the-art results of the OS/OS setup for several models (e.g., the CNN for SC; ResNet for SV) – the point here is not that the results using SS or SU are in any way superior to OS, but that they are sometimes not worse. Given this analysis, we believe to have strong evidence that the tested DNN models do not rely on SST (they might model them, but are ultimately able to produce similar results without them). We conjecture that this is due to purely modeling FBA is easy and efficient enough: It quickly brings the loss down to a local minimum from which the training cannot recover as switching features to a different set would introduce too high intermediate losses again. This is in accordance with studies claiming that DNNs are lazy in learning complex concepts when easy ones work already [31], and tend to learn shortcuts [32]. To put it drastically, DNNs "cheat" by ignoring the harder task of modeling dynamics if they can.

Technically speaking, the model overfits to FBA information to the point of neglecting other evidence. An ablation study gives further evidence of this: Remarkably, replacing the CNN in the TIMIT SV task with a simplistic model per speaker that only holds a row-wise average of all the speaker's training spectrograms (i.e., a single-Gaussian model with unit variance) outperforms the CNN. Apparently, having access to the full frequency resolution of the spectrogram at decision time is a larger advantage to the simple model than pooling in frequency *and time* is for the CNN. Overcoming this problem calls for a novel form of regularization on the task-level, which will be explored in the following section.

3. Regularization approaches to increase SST exploitation

3.1. Objective and context

Our objective here is to find ways to force models to revert to SST. As one implication of the results of Section 2 is that the predicted performance improvements [10] from exploiting SST are still to be harvested, this holds large prospects. It could be realized, e.g., by combining a model that focuses on FBA with one regularized to focus on SST [33].

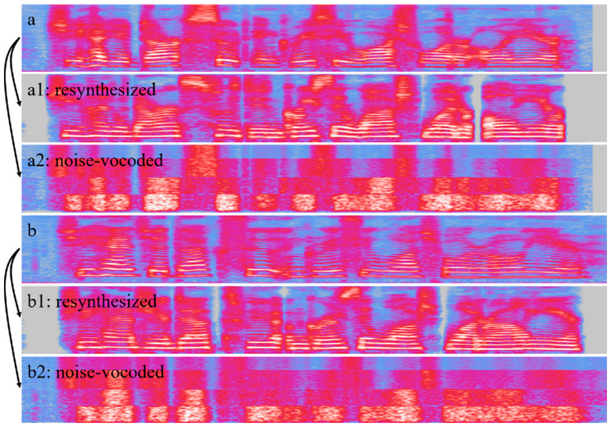

Fig. 2. Visualization of FBA equalization: Compressed spectrograms of the same sentence (SA1) of a male (a: MDAB0) and female (b: FCJF0) TIMIT speaker with derived synthesized (a1/b1) and noise-vocoded (a2/b2) variants.

3.2. Methodology and experimental setup

First, we test the conjecture that DNNs only model those features necessary to solve the task, starting with the easiest to learn. Therefor, we increase the difficulty of the task by including an acoustically more challenging dataset and check for an increase of actual SST modeling via the test established before, using the VoxCeleb corpus while keeping other parameters equal to the TIMIT analysis in Section 2 (i.e., for the CNN, RNN, and ResNet models we keep the experimental setup compatible with [13] while for the F-ResNet model we keep it compatible with [29]). VoxCeleb1 contains 148,642 utterances of varying length (few seconds up to several minutes) from 1,211 different speakers recorded in the wild, thus holding a variety of background noises including cross talk; VoxCeleb2 contains more than 1 million utterances from 5,994 speakers. We use the standard experimental protocol of Chung et al. to train a SV system on VoxCeleb2 , and evaluate SV performance on the "hard" test set of VoxCeleb1 [29]. We omit SC results on VoxCeleb as experiments in Section 2 led to similar conclusions for both tasks.

Second, we aim at actively nudging models to learn SST by making FBA less attractive, inspired by the previous approach that scrambled SST. While scrambling the frequency axis by naïvely randomizing the rows in a spectrogram would effectively remove SST as well (as it depends on the evolution of sound over frequency band borders), we instead propose the regularization strategy of largely *equalizing* (instead of randomizing) FBA amongst speakers while retaining their discriminative SST. Note that this is done specifically to study the effect of devalued FBA on SST-modeling by our DNNs, not as a general way to augment speech data [34]. Specifically, we chose two equalization strategies (cf. Fig. 2) on the TIMIT corpus: (a) Noise-vocoding the sentences with just 4 broad frequency bands [35], modulated by the original speakers' energy levels, to *reduce* spectral differences; and (b) using a custom version of the Slang TTS [36] speech synthesizer (able to use speaker-specific phoneme-level timing annotations and energy contours from the TIMIT corpus as additional input) to recreate every sentence from its text transcript with an identical synthetic voice, thus *eliminating* spectral differences but keeping some SST.

3.3. Results and discussion

SV results on VoxCeleb are shown in the left part of Table 3. Comparing them with SV on standard TIMIT (Table 2) and using the cell annotations marked by ⇓, the following is noteworthy: (g) On TIMIT, competitive results are achieved using SU/SU (CNN) and SS/SS (RNN, ResNet) that are only marginally worse than the best result per model and in particular strong in comparison with OS/OS. On VoxCeleb, OS/OS now clearly is better than all other combinations that include randomization by a larger margin. While the absolute scores are bad for the two simpler models, the ResNet shows a reasonable EER using OS/OS on VoxCeleb, only about 3 times worse than the best result on the much simpler TIMIT and about twice as good as the best result involving any randomization. The F-ResNet shows state-of-the-art performance under OS/OS (i.e., normal) conditions, thereby conforming that the experimental setup and codebase used for these experiments is sound. We conclude that providing a more challenging task (one that cannot be solved relying on simpler frequency-domain features alone, as research prior to the deep learning era has shown [37]) stimulates the exploitation of SST in models to some degree, depending on the ability of the model — as all conditions with scrambled SST fall far behind in performance (by a factor of ≥ 2.75 for the best model, F-ResNet).

The middle part of Table 3 contains the results on noise-vocoded TIMIT, where speakers' individual timbre has been largely removed. Again comparing with the respective results on standard TIMIT, it is noteworthy that (h) best results are now achieved by OS/OS for the RNN and ResNet. However, the CNN still has best results involving random timing, and also for the RNN and ResNet models, the margin for OS/OS is small and EERs are 4-5 times higher than on standard TIMIT. We conclude that the effect seen on VoxCeleb (a harder task makes the models start learning SST) is visible to some degree, but less pronounced. (i) The same is true for the results on resynthesized TIMIT (cf. right part of Table 3), except that best results for the CNN are achieved using another form of randomization and the EERs are 8-11 times worse here. (k) It is different for the F-ResNet, where best results on noise-vocoded TIMIT are still achieved using SS/SS (and only with a very thin margin using OS/OS on resynthesized TIMIT). We conjecture that this is again due to the tininess of the TIMIT database for training this large model (see (j) discussed in Section 2).

We conclude that diminishing the dominance (i.e., speaker discriminativeness) of FBA partially (using noise vocoding) or fully (through resynthesis) brings forth some exploitation of SST for SV, but not optimally (which would be evidenced not necessarily by lower absolute EERs, but by larger margins between top OS/OS results and everything else). Moreover, such an effect is even less pronounced when switching tasks from SV to SC (tables omitted): No benefit of OS/OS can be observed on TIMIT-NV and only little evidence for it is seen on TIMIT-Syn. We conjecture that the attempt to model SST apart from FBE is suboptimal (in accordance with the literature [6,7] that categorizes SST as of subordinate importance but helpful in addition to FBA). Evidence for this is the experiment on VoxCeleb that shows that under challenging circumstances ResNets can achieve unparalleled results by exploiting FBA and SST jointly.

4. Conclusions, future work and limitations

In this paper, we have presented the first systematic study on learning supra-segmental temporal features by DNNs for SR. Not focusing on presenting a new kind of model or SR methodology, we have instead shown that state-of-the-art CNN, RNN and ResNet models for SV and SC, when trained on clean data, simply ignore any useful supra-segmental temporal cues in the audio signal despite contrary conjectures in the literature and the models' principal abilities to learn such features. We have called this phenomenon “deep cheating”. It is relevant since related work provided evidence [10] that improved modeling of such higher-level features should result in one order of magnitude lower error rates in related tasks and hence holds a key for targeted future research, guided by our test to quantify actual SST exploitation. It is also of importance in the context of explaining *how* DNNs achieve their superior results (XAI), where our explanation goes beyond activation visualization to explain individual classifications towards a broader understanding of signal processing by DNNs.

Furthermore, we have presented two approaches to force DNNs to exploit SST, and measured their effectiveness: (a) Increasing task difficulty by using acoustically more challenging data (VoxCeleb instead of TIMIT), and (b) removing the discriminative power of FBA by equalizing speakers' timbre. The results indicate that both approaches achieve respective results nominally, thereby confirming other studies that attest DNNs laziness in modeling only the easiest available features to solve a given task. Theoretical and empirical studies suggest that scaling up training time might help overcome such imperfect local minima [38].

We have conducted extensive experiments to verify the correctness and stability of our results for a wide range of design choices. Our claims hold for reasonable settings of the hyperparameters learning rate, number of epochs, segment and hop length, and embedding size; when using the original loss functions of published models instead of speed-ups; with varying number of frequency bands for the noise vocoder or using MBROLA [39] as another synthesizer; and for evaluating on VoxCeleb2. TIMIT, though small, is a sound basis for our findings (cf. [38]): It has been used successfully for this purpose by the community before; we do not observe problems with overfitting for all but the F-ResNet model, although by construction of the mini batches, we only exploit a fraction of the available training data; it contains pure voices without exogenous difficulties (noise, brevity, …), offering to study SR capability in isolation and hence granting an unbiased look at DNNs' abilities for voice modeling.

However, we have also shown that attempting to learn SST apart from FBA results in severely underperforming models that verge on random SV results and are even less helpful for SC. Hence, our results are preliminary with respect to finding better ways of exploiting speaker-specific SST with DNNs. Thinking of perceivable instantiations of SST like personal linguistic melody, it is elusive how such strong rhythmic-prosodic patterns are not picked up by any of the most capable general pattern recognition methods we know today, deep neural networks. Future work should therefor concentrate on finding inductive biases for deep networks that fully exploit the time axis for speaker specificity. Inspired by how auto-regressive self-supervised learning in text processing works [40] and building on the success of respective large language models, transformer-based architectures and large-scale pre-training could be a way to integrate handling of dynamic features either directly into the model or into methods for speech augmentation. Such work should use the test presented in Section 2 to benchmark the success of actually exploiting STT. Our code is available online at https://tinyurl.com/deepcheating.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We used well-known and available benchmark datasets for all experiments. We made all our code available (link in manuscript)

Acknowledgments

The authors thank Bernd Freisleben, Claude Olivier Lehmann, Frank-Peter Schilling and Richard Hahnloser for encouraging discussions, and their collaborators on voice recognition methodology over the years: Oliver Dürr, Yanick X. Lukic, Carlo Vogt, Feliks Hibraj, Sebastiano Vascon, Marcello Pelillo, Patrick Gerber, Sebastian Glinski-Haefeli, Benjamin Neusser, Savin Niederer, Timon Gygax, Jörg Egli, Niclas Simmler, Amin Trabi, Jan Sonderegger, Patrick Walter, Christian Lauener, Pascal Sager, and Sydney M. Nguyen.

References

[1] A. Brown, J. Huh, J.S. Chung, A. Nagrani, D. Garcia-Romero, A. Zisserman, VoxSRC 2021: The third VoxCeleb speaker recognition challenge, 2022, arXiv:2201.04583.

[2] L. Tuggener, J. Schmidhuber, T. Stadelmann, Is it enough to optimize CNN architectures on ImageNet? Front. Comput. Sci. (2022).

[3] H.W. Lin, M. Tegmark, D. Rolnick, Why does deep and cheap learning work so well? J. Stat. Phys. 168 (6) (2017) 1223–1247.

[4] T.J. Sejnowski, The unreasonable effectiveness of deep learning in artificial intelligence, Proc. Natl. Acad. Sci. USA 117 (48) (2020) 30033–30038.

[5] M. Ivanovs, R. Kadikis, K. Ozols, Perturbation-based methods for explaining deep neural networks: A survey, Pattern Recognit. Lett. 150 (2021) 228–234.

[6] P. Rose, Forensic Speaker Identification, Taylor & Francis, London and New York, 2002.

[7] J.H. Hansen, T. Hasan, Speaker recognition by machines and humans: A tutorial review, IEEE Signal Process. Mag. 32 (6) (2015) 74–99.

[8] A. Leemann, M.-J. Kolly, V. Dellwo, Speaker-individuality in suprasegmental temporal features: Implications for forensic voice comparison, Forensic Sci. Int. 238 (2014) 59–67.

[9] S. Shahnawazuddin, N. Adiga, H.K. Kathania, B.T. Sai, Creating speaker independent ASR system through prosody modification based data augmentation, Pattern Recognit. Lett. 131 (2020) 213–218.

[10] T. Stadelmann, B. Freisleben, Unfolding speaker clustering potential: A biomimetic approach, in: Proc. ACM MM, 2009, pp. 185–194.

[11] Y.X. Lukic, C. Vogt, O. Dürr, T. Stadelmann, Speaker identification and clustering using convolutional neural networks, in: Proc. MLSP, 2016, pp. 1–6.

[12] Y.X. Lukic, C. Vogt, O. Dürr, T. Stadelmann, Learning embeddings for speaker clustering based on voice equality, in: Proc. MLSP, 2017, pp. 1–6.

[13] T. Stadelmann, S. Glinski-Haefeli, P. Gerber, O. Dürr, Capturing suprasegmental features of a voice with rnns for improved speaker clustering, in: Proc. ANNPR, 2018, pp. 333–345.

[14] Y. Zhao, T. Zhou, Z. Chen, J. Wu, Improving deep CNN networks with long temporal context for text-independent speaker verification, in: Proc. ICASSP, 2020, pp. 6834–6838.

[15] F. Ye, J. Yang, A deep neural network model for speaker identification, Appl. Sci. 11 (8) (2021) 3603.

[16] A. Graves, A.-R. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: Proc. ICASSP, 2013, pp. 6645–6649.

[17] W.M. Fisher, The DARPA speech recognition research database: Specifications and status, in: Proc. DARPA Workshop on Speech Recognition, 1986, pp. 93–99.

[18] A. Nagrani, J.S. Chung, A. Zisserman, VoxCeleb: A large-scale speaker identification dataset, in: Proc. Interspeech, 2017, pp. 2616–2620.

[19] S.B. Davis, P. Mermelstein, Evaluation of acoustic parameters for monosyllabic word identification, J. Acoust. Soc. Am. 64 (S1) (1978) S180–S181.

[20] S. Soleymani, A. Dabouei, S.M. Iranmanesh, H. Kazemi, J. Dawson, N.M. Nasrabadi, Prosodic-enhanced siamese convolutional neural networks for cross-device text-independent speaker verification, in: Proc. BTAS, 2018, pp. 1–7.

[21] Z. Bai, X.-L. Zhang, Speaker recognition based on deep learning: An overview, Neural Netw. 140 (2021) 65–99, http://dx.doi.org/10.1016/j.neunet.2021.03.004.

[22] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. CVPR, 2016, pp. 770–778.

[23] T. Stadelmann, V. Tolkachev, B. Sick, J. Stampfli, O. Dürr, Beyond ImageNet: Deep learning in industrial practice, in: Applied Data Science, Springer, 2019, pp. 205–232.

[24] D. Snyder, D. Garcia-Romero, D. Povey, S. Khudanpur, Deep neural network embeddings for text-independent speaker verification, in: Proc. Interspeech, 2017, pp. 999–1003.

[25] H.N. Pinheiro, T.I. Ren, A.G. Adami, G.D. Cavalcanti, Variational dnn embeddings for text-independent speaker verification, Pattern Recognit. Lett. 148 (2021) 100–106.

[26] Y. Wu, C. Guo, H. Gao, X. Hou, J. Xu, Vector-based attentive pooling for text-independent speaker verification, in: Proc. Interspeech, 2020, pp. 936–940.

[27] W. Xie, A. Nagrani, J.S. Chung, A. Zisserman, Utterance-level aggregation for speaker recognition in the wild, in: Proc. ICASSP, 2019, pp. 5791–5795.

[28] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, W. Liu, CosFace: Large margin cosine loss for deep face recognition, in: Proc. CVPR, 2018, pp. 5265–5274.

[29] J.S. Chung, J. Huh, S. Mun, M. Lee, H.S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, I. Han, In defence of metric learning for speaker recognition, in: Proc. Interspeech, 2020, pp. 2977–2981.

[30] T. Stadelmann, B. Freisleben, Dimension-decoupled gaussian mixture model for short utterance speaker recognition, in: Proc. ICPR, 2010, pp. 1602–1605.

[31] T. Mansour, Deep Neural Networks are Lazy: On the Inductive Bias of Deep Learning (Master's thesis), MIT, 2019.

[32] A.J. DeGrave, J.D. Janizek, S.-I. Lee, AI for radiographic COVID-19 detection selects shortcuts over signal, Nat. Mach. Intell. 3 (7) (2021) 610–619.

[33] M. Ganaie, M. Hu, A. Malik, M. Tanveer, P. Suganthan, Ensemble deep learning: A review, Eng. Appl. Artif. (2022).

[34] D.S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E.D. Cubuk, Q.V. Le, SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition, in: Proc. Interspeech, 2019, pp. 2613–2617, http://dx.doi.org/10.21437/Interspeech.2019-2680.

[35] R.V. Shannon, F.-G. Zeng, V. Kamath, J. Wygonski, M. Ekelid, Speech recognition with primarily temporal cues, Science 270 (5234) (1995) 303–304.

[36] SlowSoft GmbH, Slang TTS speech synthesizer, 2021, https://slowsoft.ch/eng/products.html. (Accessed: 15 Feb 2022).

[37] T. Stadelmann, Voice Modeling Methods for Automatic Speaker Recognition, (Ph.D. thesis), Philipps-Universität Marburg, 2010.

[38] A. Power, Y. Burda, H. Edwards, I. Babuschkin, V. Misra, Grokking: Generalization beyond overfitting on small algorithmic datasets, 2022, arXiv preprint arXiv:2201.02177.

[39] T. Dutoit, V. Pagel, N. Pierret, F. Bataille, O. Van der Vrecken, The MBROLA project: Towards a set of high quality speech synthesizers free of use for non commercial purposes, in: Proc. ICSLP, 1996, pp. 1393–1396.

[40] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: Proc. ICLR, 2013.