GENERALISED DISCRIMINATIVE TRANSFORM VIA CURRICULUM LEARNING FOR SPEAKER RECOGNITION

Erik Marchi, Stephen Shum, Kyuyeon Hwang, Sachin Kajarekar, Siddharth Sigtia, Hywel Richards, Rob Haynes, Yoon Kim, John Bridle

Siri Speech, Apple Inc.

ABSTRACT

In this paper we introduce a speaker verification system deployed on mobile devices that can be used to personalise a keyword spotter. We describe a baseline DNN system that maps an utterance to a speaker embedding, which is used to measure speaker differences via cosine similarity. We then introduce an architectural modification which uses an LSTM system where the parameters are optimised via a curriculum learning procedure to reduce the detection error and improve its generalisability across various conditions. Experiments on our internal datasets show that the proposed approach outperforms the DNN baseline system and yields a relative EER reduction of 30–70% on both text-dependent and text-independent tasks under a variety of acoustic conditions.

Index Terms— Speaker Verification, Deep Learning, LSTM, Curriculum Learning

1. INTRODUCTION

Speaker verification aims to confirm the identity of a speaker by matching some representation of an incoming test phrase to that of a set of speaker-dependent enrolment phrases. At Apple, we are interested in personalising the *Hey Siri* detector by verifying the speaker's voice before triggering Siri; without this, the always-on detector described in [1] would respond whenever anyone in the vicinity says the trigger phrase (or even something else that sounds similar). To reduce the inconvenience of such false triggers, this feature introduces an optional enrolment session, during which the user says five phrases, each of which begin with 'Hey Siri.' Enrolment reduces not only the probability that 'Hey Siri' spoken by another person will trigger the user's iPhone, but also the rate at which similar-sounding phrases might trigger Siri.

In the last three years, we have seen approaches to speaker verification shift from i-vectors and PLDA [2, 3, 4] to obtaining speaker representations using DNNs, LSTMs, and CNNs. In particular, Heigold *et al.* [5] proposed an end-to-end text-dependent speaker verification system that derives a 'd-vector' from a DNN and an LSTM. Subsequently, Zhang *et al.* [6] applied an attention mechanism to a text-dependent end-to-end system by jointly optimising a deep CNN and an attention model learned from a phonetic representation. On the text-independent front, Snyder *et al.* [7] introduced an end-to-end system using a DNN architecture that maps a variable-length speech segment to a fixed-dimensional speaker embedding. Finally, another recent study proposed a deeper end-to-end architecture (>20M parameters) using a residual CNN and gated recurrent units that worked for both text-dependent and text-independent tasks [8].

Except for the last study mentioned, most investigations into speaker representations focus on either text-dependent or text-independent scenarios individually. At Apple, however, our data is more integrated. The 'Hey Siri' requests we see come in one of two forms: (a) just the 'Hey Siri' trigger; or (b) the trigger followed by the payload request (e.g., 'Hey Siri, how's the weather today?'). Given the extra speech, it seems reasonable to expect a more reliable speaker representation from (b). To the best of our knowledge, no study has looked at extending a text-dependent speaker representation towards a more text-independent (or *less-constrained text*) scenario. Furthermore, given the computational constraints of battery-powered devices, our ultimate goal is to deploy a single discriminative transform that can perform speaker verification on both text-dependent and text-independent tasks.

In this paper, we first present our baseline, text-dependent model for speaker verification that uses a DNN to extract speaker-specific information from a supervector-based representation of the trigger phrase. We then propose an LSTM model together with two learning strategies aimed at improving generalisation: curriculum learning [9] and multi-style training. While multi-style training is a commonly used technique in speaker recognition [10], curriculum learning has never been applied in the field, even though it has been used successfully for image classification [11], automatic speech recognition [12], and handwriting text recognition [13]. We show that the proposed modifications yield a system that can halve the equal error rate achieved by the baseline system and handle utterances containing less-constrained text. The remainder of the paper is organised as follows. Section 2 describes the baseline DNN system and outlines the LSTM system. Curriculum learning and multi-style training strategies are discussed in Section 3. Section 4 introduces the datasets and elaborates on the results before we conclude in Section 5.

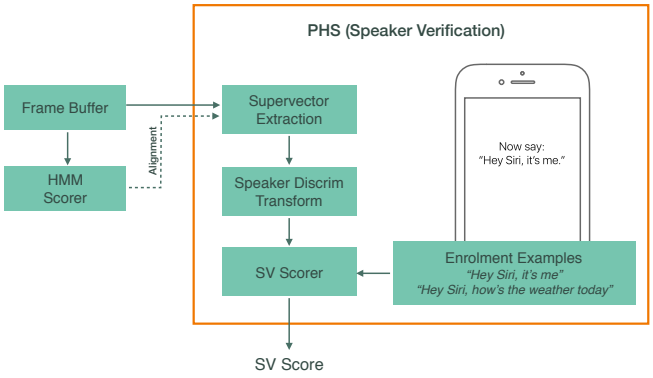

Fig. 1: Overview of the PHS system. Note that the frame buffer contains the acoustic features, and the HMM scorer is only part of the actual detector [1] which for the sake of clarity is not entirely depicted here.

2. SYSTEM OVERVIEW

Figure 1 shows a high-level diagram of the personalised *Hey Siri* system (PHS). We first extract 26 MFCCs using a window size of 25 ms at a rate of 100 frames per second. Then, we derive a speaker supervector from the *Hey Siri* detector by concatenating state segment means from the output of the dynamic programming accumulator of the HMM scorer [1], resulting in 442-dimensional vector. Note that the dynamic programming accumulator has 28 states, however the first 10 states model silence and the eleventh state models the start of the /h/, so we choose not to use them in our supervector. A specially-trained DNN transforms the supervector into a 'speaker space' where, by design, patterns from the same speaker tend to be close, whereas patterns from various speakers tend to be further apart. The speaker verification (SV) score is computed in the 'speaker space' where the cosine similarities to the reference patterns created during enrolment are averaged as follows:

$$SV_{score}(u_a, spk) = \frac{1}{N}\sum_{i=1}^{N}\frac{f_{nn}(u_a)^\top f_{nn}(u_i^{spk})}{\|f_{nn}(u_a)\|\|f_{nn}(u_i^{spk})\|} \quad (1)$$

where u_a is the input vector we want to test, u_i^{spk} is the i^{th} enrolment reference vector for speaker spk, N is the total number of enrolments, and f_{nn} is the speaker discriminative transform. The score is then compared with a threshold to decide whether the sound that triggered the detector is likely to be 'Hey Siri' spoken by the enrolled user.

2.1. Speaker Discriminative Transform

For a speaker discriminative transform we first consider a fully-connected multilayer neural network with sigmoid nonlinearities, followed by a fully-connected linear layer, and a softmax layer [14] with K units, where K is the number of speakers in the transform training set (cf. Section 4). Given a labelled training sample $(u, y) \in X$ where X is the training set, u is a input vector, and y is the target speaker index, we train the model to minimise the negative log-likelihood (cross-entropy) of the softmax distribution: $\ell((u, y); \Theta) = -\log\left[e^{z_j} / \sum_{k=1}^{K} e^{z_k}\right]$, where Θ are the neural network parameters, j is the target speaker, z are the unnormalised log probabilities predicted by the linear transformation comprised in the softmax layer: $z_j = w_j^\top h + b_j$, where w_j and b_j are the weights and bias, and h is the vector of activations of the last hidden layer. In the evaluation phase, the softmax layer is removed and the output of the linear layer is used as a speaker embedding.

2.1.1. DNN Baseline

The DNN baseline speaker discriminative transform consists of four fully-connected sigmoidal layers followed by a 128-unit linear layer and a softmax layer with 18k units. Batch normalisation is applied at every layer except the last. The parameters were initialised from a uniform distribution, and the optimisation is performed via stochastic gradient descent. The gradient of the cost function L is computed with respect to some parameter $\theta \in \Theta$, with $L(\Theta) = \sum_{m=1}^{M} \ell((u_m, y_m); \Theta)$, where M is set to 256 and represents the number of training samples randomly selected in a mini-batch. The training is initialised at a learning rate of $1\text{e}{-4}$ with a momentum of 0.9 and a weight decay of $1\text{e}{-4}$ at the end of each epoch, where one epoch is considered to be a full sweep on the training set. We monitor the training convergence on a cross-validation set and halve the learning rate as appropriate. The described DNN system serves as a baseline reference to our previous on-device configuration. Note that the DNN system supports text-dependent task only and some of the system parameters (e.g., the HMM scorer, and the 26 MFCCs) are DNN-specific. Including more details on the optimisation of the DNN model is beyond the scope of this paper; instead, we focus on the more flexible LSTM system that demonstrates significant improvements and is flexible enough to handle less-contrained text.

2.1.2. LSTM System

The LSTM system replaces the four fully-connected sigmoidal layers of the DNN baseline with a single recurrent layer containing 512 LSTM units. While vanilla LSTMs [15] have multiple outputs at each timestep, we only connect the last LSTM output h_M (where M is the length of the input sequence u) to the fully-connected linear layer in order to obtain a single, utterance-level speaker representation. The input to the LSTM is simply the sequence of MFCC frames (20 MFCCs per frame, 25ms data window, 100 frames per second). Note that the LSTM system did not perform significantly better with 26 MFCCs input; thus, we only report results with 20 MFCCs input. For stochastic optimisation, we use Adam [16] with an initial learning rate of $1\text{e}{-3}$ and a mini-batch size of 128. We also perform 8-bit quantisation on the network parameters for every system evaluated in this paper.

3. IMPROVING GENERALISATION

To improve the generalisation of the speaker discriminative transform, we adopt and combine both multi-style training and curriculum learning. Both strategies help increase robustness under various acoustic conditions, and the latter is especially instrumental in teaching the model to handle different textual content. To help explain the details of these two paradigms, we first define the following training subsets X_k where $k \in \{\texttt{hs}, \texttt{hs+pl}, \texttt{pl}, \texttt{all}\}$ as follows:

- $X_{\texttt{hs}}$ contains samples only with the global keyword (e.g., 'Hey Siri');
- $X_{\texttt{hs+pl}}$ consists of samples with a first part containing the keyword followed by a payload part with variable text (e.g., 'Hey Siri, what time is it?');
- $X_{\texttt{pl}}$ consists of samples only containing the payload (e.g., 'what time is it?');
- $X_{\texttt{all}} = \{X_{\texttt{hs}} \cup X_{\texttt{hs+pl}} \cup X_{\texttt{pl}}\}$.

Finally, to distinguish between audio recordings in their original form and those we artificially create by adding noise and/or convolving with various room impulse responses (as described in Section 4), we denote the latter using the superscript *sim* (e.g., $X^{sim}_{\texttt{all}}$).

In **multi-style training** (MST), we train with an augmented data set that contains both the original and artificially created versions of the training set; i.e., $\{X_k \cup X^{sim}_k\}$. The same principle applies to improving generalisation to obtain a text-independent system where the various subsets are all merged together to create training set X_{all}.

The **curriculum learning** (CL) paradigm formalises a general principle of learning simpler concepts first before gradually learning more complex ones. This adheres to our desire to build a text-independent speaker verification system by learning first a text-dependent task that uses a known, fixed phrase and then learning to handle a more complex task that contains less-constrained text content. In particular, the network starts by learning a discriminative transform on a text-dependent task from samples containing only the keyword ($X_{\texttt{hs}}$). The complexity of the task is then increased slightly by introducing samples that contain a payload in addition to the original trigger ($X_{\texttt{hs+pl}}$), and so on. This paradigm can be similarly applied to learn from various acoustic scenarios by training first on clean data and then running an additional learning step on the augmented data (i.e., CL0 in Table 1).

Table 1 presents curricula of varying complexities that we consider in this paper. The first three – VAN, MST0, and CL0 – demonstrate the impact of both multi-style training and curriculum learning on a system that focuses solely on the 'Hey Siri' trigger. The next one – MST1 – is a naive baseline, of sorts, in which all the data of all types is fed into the model at once. Finally, the last two – CL1 and CL2 – outline the effects of curriculum learning as a way to generalise towards the ability to handle text-independent tasks. Note that a learning step consists of a full sweep on the training set and that the first CL stage is always performed on $X_{\texttt{hs}}$ until convergence on a cross-validation sets.

	Curriculum Learning Steps
VAN	$X_{\texttt{hs}}$
MST0	$\{X_{\texttt{hs}} \cup X^{sim}_{\texttt{hs}}\}$
CL0	$X_{\texttt{hs}} \rightarrow \{X_{\texttt{hs}} \cup X^{sim}_{\texttt{hs}}\}$
MST1	$\{X_{\texttt{all}} \cup X^{sim}_{\texttt{all}}\}$
CL1	$X_{\texttt{hs}} \rightarrow X_{\texttt{hs+pl}} \rightarrow X_{\texttt{pl}} \rightarrow X_{\texttt{all}}$
CL2	$X_{\texttt{hs}} \rightarrow X_{\texttt{hs+pl}} \rightarrow X_{\texttt{pl}} \rightarrow X_{\texttt{all}} \rightarrow \{X_{\texttt{all}} \cup X^{sim}_{\texttt{all}}\}$

Table 1: Different curricula types. Datasets are either merged ($\cup$), or used at sequential training steps ($\rightarrow$). VAN denotes 'vanilla' data in its original form; and as discussed in Section 3, MST and CL stand for 'multi-style training' and 'curriculum learning', respectively.

		#utterances	#speakers	#utt/spk ($\tilde{m}$)
Train ($X_{\cdot}$)		2.5M	18k	>20 (118)
Train ($X^{sim}_{\cdot}$)		1.5M	18k	>20 (118)
iP-van	enrol	2.5k	500	>4 (5)
	test	53k	500	>40 (106)
FF	enrol	490	98	>4 (5)
	test	11k	102	>20 (118)
iP-sim	enrol	1.3k	250	>4 (5)
	test	19.5k	245	>1 (70)

Table 2: Statistics for the training sets ($X_{\cdot}$, and $X^{sim}_{\cdot}$), and for the evaluation sets with details on the utterances used to 'enrol' and 'test' our models. $\tilde{m}$ is the median of the number of utterances per speaker.

We did investigate a fairly exhaustive set of curriculum learning step permutations, including starting with either $X_{\texttt{hs+pl}}$ or $X_{\texttt{pl}}$ and introducing X^{sim}_k at an earlier stage. However, CL1 and CL2 not only gave the best results, but they also consistently converged fastest during training. It seems as though learning on the shorter utterances in $X_{\texttt{hs}}$ can really help jumpstart the model's ability to discriminate between speakers. More experiments are required to substantiate these findings but are beyond the scope of this paper – we look forward to investigating this in future work.

4. DATASETS AND EXPERIMENTS

Statistics about the data used for training and evaluation are shown on Table 2. In general, we want to demonstrate robust performance on the following two evaluation sets:

- Vanilla iPhone (iP-van) data, consisting of 'Hey Siri' requests sent to our servers from production iPhones;
- Far-field (FF) data, consisting of 'Hey Siri' requests from various distances (6ft, 9ft, 12ft, 15ft) recorded in various rooms of 10 different houses.

However, while our original training material ('Train ($X_{\cdot}$)' in Table 2) resembles that of iP-van, it is a significant mismatch from the FF data. To help compensate for this mismatch, we

		#params	EER [%] iP-van	FF	iP-sim
1	DNN (VAN)	0.4M	3.99	11.80	5.26
2	LSTM (VAN)	1.2M	3.10	3.52	5.01
3	DNN (MST0)	0.4M	4.20	5.00	4.52
4	LSTM (MST0)	1.2M	3.07	3.43	**3.52**
5	LSTM (CL0)	1.2M	**2.97**	**3.21**	**3.52**

Table 3: Performance comparison across different architectures and training strategies on text-dependent test sets.

also create a simulated training set ('Train ($X_{\cdot}^{sim}$)' in Table 2) that is artificially generated from the original 'Train ($X_{\cdot}$)' data by convolving a subset of the utterances with one of 500 different room impulse responses and, optionally, adding one of two car noise types at various SNR levels. In light of this, we also provide results on a comparable evaluation set:

- Simulated iPhone (iP-sim) data, which we artificially generate from the iP-van evaluation data.

4.1. Text-dependent: 'Hey Siri'

We first present results obtained by training and evaluating only on fixed-text data and then elaborate on the results obtained using less-constrained text material. Results are reported in terms of equal error rate (EER) with t-norm score normalisation [17]. Comparing results between Rows 1 and 2 of Table 3 show that the LSTM system achieves a lower detection error than the DNN baseline on all three evaluation sets leading to 23%, 71%, and 5% relative EER reduction on iP-van, FF, and iP-sim datasets, respectively. Applying MST0 yields a dramatic gain on the FF data for the DNN (Rows 1 and 3), while the LSTM sees significant EER improvements only on the simulated evaluation data (Rows 2 and 4). Lastly, we see a slight improvement on the iP-van and FF sets using CL0 instead of just MST0 for LSTM training (Rows 4 and 5). Although it did not further improve the iP-sim result, it does seem to support the conjecture that curriculum learning can help improve robustness under various acoustic conditions.

4.2. Text-independent: 'Hey Siri, what time is it?'

Figure 2 reports results obtained at different curriculum learning stages. For clarity, we only report results obtained with the LSTM, as it consistently outperforms the DNN system. The left-most element on the horizontal axis (VAN) denotes a vanilla system trained using just the fixed-text data. Given that the network has never seen any `hs+pl` data in training, it is not surprising that its result on the less-constrained `hs+pl` data is significantly worse than on the fixed-text `hs` data. CL1 denotes a model that went through the four learning steps as indicated in Table 1. We can see that it is able to achieve a much lower EER on both the `hs+pl` and `hs` evaluations. In fact, we observe a relative reduction of 54% on iP-van and 70% on FF, obtaining a speaker transform that is able to extract relevant speaker representations also from the entire utterance including the payload. Additionally, it confirms that if

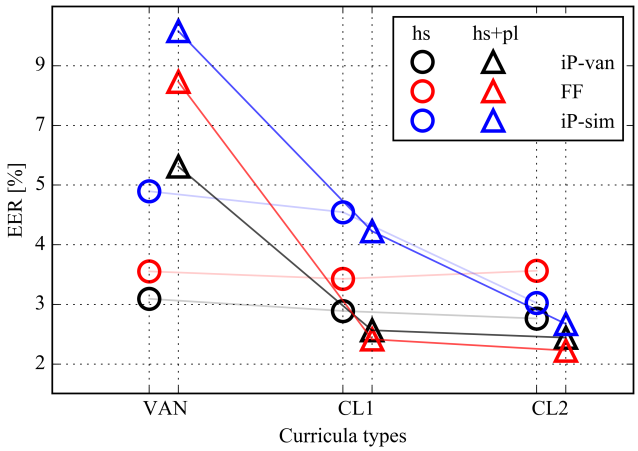

Fig. 2: Performance comparison across different stages of the curriculum learning procedure for the LSTM system. Circles represent evaluation sets containing just `hs` data, while triangles represent evaluation sets containing `hs+pl` data.

		EER [%] iP-van	FF	iP-sim
`hs`	MST1	3.09	3.94	3.08
	CL2	**2.83**	**3.53**	**2.80**
`hs+pl`	MST1	2.92	3.23	2.88
	CL2	**2.59**	**2.44**	**2.76**

Table 4: Performance comparison between MST1 and CL2.

we have access to `hs+pl` we can further decrease the detection error compared with just using a text-dependent system. If we then include a last curriculum learning stage with augmented data, as the CL2 system does, we get a further EER reduction on all evaluation sets (down to 2.59%, 2.44%, and 2.76% EER on iP-van, FF, and iP-sim, respectively).

As a final sanity check, Table 4 shows the results of a system trained using MST1, which produces a model that has seen as much data as the model produced by CL2. However, we can see that CL2 consistently outperforms MST1 at every point of comparison. Furthermore, because it had so much more data to handle, the MST1 model also took substantially longer to train and converge.

5. CONCLUSIONS

The system described here, when trained following a curriculum learning procedure, improves both the robustness against various acoustic conditions and the generalisability towards less constrained-text scenarios. Compared to the DNN baseline system, the proposed speaker discriminative system yields a relative EER reduction of 30–70% on both text-dependent and text-independent conditions while keeping the network size small enough to be deployed on device. Future work may focus towards automated curriculum learning [18] where a teacher network automatically selects which tasks the student network has to learn [19].

6. REFERENCES

[1] Siri Team, "Hey Siri: An on-device DNN-powered voice trigger for Apple's personal assistant," *Apple Machine Learning Journal*, vol. 1, issue 6, October 2017.

[2] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1, pp. 19–41, Jan. 2000.

[3] Najim Dehak, Patrick J. Kenny, Reda 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, May 2011.

[4] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Phonetically-constrained PLDA modeling for text-dependent speaker verification with multiple short utterances," in *Proceedings 38th IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, Vancouver, Canada, 2013, pp. 7673–7677, IEEE.

[5] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, Shanghai, China, March 2016, pp. 5115–5119, IEEE.

[6] S. X. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, "End-to-end attention based text-dependent speaker verification," in *Proceedings IEEE Spoken Language Technology Workshop, SLT 2016*, San Diego, California, December 2016, pp. 171–178, IEEE.

[7] David Snyder, Pegah Ghahremani, Daniel Povey, Daniel Garcia-Romero, Yishay Carmiel, and Sanjeev Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Proceedings of the IEEE Spoken Language Technologies (SLT) Workshop 2016*. 2016, pp. 165–170, IEEE.

[8] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *CoRR*, vol. abs/1705.02304, pp. 8, 2017.

[9] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston, "Curriculum learning," in *Proceedings 26th Annual International Conference on Machine Learning*, New York, NY, USA, 2009, ICML '09, pp. 41–48, ACM.

[10] D. Garcia-Romero, X. Zhou, and C. Y. Espy-Wilson, "Multicondition training of gaussian plda models in i-vector space for noise and reverberation robust speaker recognition," in *Proceedgins 37th IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*. March 2012, pp. 4257–4260, IEEE.

[11] C. Gong, D. Tao, S. J. Maybank, W. Liu, G. Kang, and J. Yang, "Multi-modal curriculum learning for semi-supervised image classification," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3249–3260, July 2016.

[12] Stefan Braun, Daniel Neil, and Shih-Chii Liu, "A curriculum learning method for improved noise robustness in automatic speech recognition," *CoRR*, vol. abs/1606.06864, pp. 5, 2016.

[13] Jérôme Louradour and Christopher Kermorvant, "Curriculum learning for handwritten text line recognition," *CoRR*, vol. abs/1312.1737, 2013.

[14] John S. Bridle, *Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition*, pp. 227–236, Springer Berlin Heidelberg, Berlin, Heidelberg, 1990.

[15] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with LSTM," in *Proceedings 9th International Conference on Artificial Neural Networks, ICANN 99*, 1999, vol. 2, pp. 850–855 vol.2.

[16] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[17] Roland Auckenthaler, Michael Carey, and Harvey Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 42–54, 2000.

[18] Alex Graves, Marc G. Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu, "Automated curriculum learning for neural networks," *CoRR*, vol. abs/1704.03003, pp. 10, 2017.

[19] Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman, "Teacher-student curriculum learning," *CoRR*, vol. abs/1707.00183, pp. 14, 2017.