

ROBUST AND FINE-GRAINED PROSODY CONTROL OF END-TO-END SPEECH SYNTHESIS

Younggun Lee, Taesu Kim

Neosapience, Inc., Seoul, Republic of Korea

ABSTRACT

We propose prosody embeddings for emotional and expressive speech synthesis networks. The proposed methods introduce temporal structures in the embedding networks, thus enabling fine-grained control of the speaking style of the synthesized speech. The temporal structures can be designed either on the speech side or the text side, leading to different control resolutions in time. The prosody embedding networks are plugged into end-to-end speech synthesis networks and trained without any other supervision except for the target speech for synthesizing. It is demonstrated that the prosody embedding networks learned to extract prosodic features. By adjusting the learned prosody features, we could change the pitch and amplitude of the synthesized speech both at the frame level and the phoneme level. We also introduce the temporal normalization of prosody embeddings, which shows better robustness against speaker perturbations during prosody transfer tasks.

Index Terms— Prosody, Speech style, Speech synthesis, Text-to-speech

1. INTRODUCTION

Since Tacotron [1] paved the way for end-to-end Text-To-Speech (TTS) using neural networks, researchers have attempted to generate more naturally sounding speech by conditioning a TTS model via speaker and prosody embedding [2, 3, 4, 5, 6]. (We use the term *prosody* as defined in earlier work [4] henceforth.) Because there is no available label for prosody, learning to control prosody in TTS is a difficult problem to tackle. Recent approaches learn to extract prosody embedding from reference speech in an unsupervised manner and use prosody embedding to control the speech style [4, 5]. These models have demonstrated ability to generate speech with expressive styles with Tacotron [1] using prosody embedding. They can also transfer the prosody of one speaker to another using a different speaker ID while leaving the prosody embedding unchanged. However, we observed two limitations with the above models.

First, controlling the prosody at a specific moment of generated speech is not clear. Earlier works focused on prosody embedding with a fixed length (a length of 1 in their experiments) regardless of the length of the reference speech or that of the text input. A loss of temporal information when squeezing reference speech into a fixed length embedding is highly likely. Therefore, fine-grained control of prosody at a specific moment of speech is difficult for embedding with a fixed length. For example, we can set the global style as "lively" or "sad," but we cannot control the prosody of a specific moment with fixed-length embedding. Because humans are sensitive to subtle changes of nuance, it is important to ensure fine-grained control of prosody to represent one's intentions precisely.

Secondly, inter-speaker prosody transfer is not robust if the difference between the pitch range of the source speaker and the target

speaker is significant. For example, when the source speaker (female) has higher pitch than the target speaker (male), the prosody-transferred speech tends to show a higher pitch than the usual pitch of the target speaker.

In this work, we focus on solving these two problems. We will introduce two types of variable-length prosody embedding which have the same length as the reference speech or input text to enable sequential control of prosody. In addition, we will show that normalizing prosody embedding helps to maintain the robustness of prosody transfers against speaker perturbations. With our methods, speaker-normalized variable-length prosody embedding was able to not only to control prosody at each specific frame, but also to transfer prosody between two speakers, even in a singing voice.

2. RELATED WORK

Prosody modeling had been done in a supervised manner by using annotated labels, such as those in ToBI [7]. Problems were reported about hand annotations, and the cost was high [8].

Skerry-Ryan et al. used convolutional neural networks and a Gated Recurrent Unit (GRU) [9] to compress the prosody of the reference speech [4]. The output, denoted by p , is fixed-length prosody embedding. They enabled prosody transfers using the prosody embedding, but they could not gain control of prosody at a specific point of time. Another problem was also reported [5]; fixed-length prosody embedding worked poorly if the length of the reference speech was shorter than the speech to generate. In addition, variable-length prosody embedding was also implemented using the output of the GRU at every time step [4]. However, this method did not draw attention because it could not obtain satisfactory results given that it was not robust with regard to text and speaker perturbations. We noted the usefulness of variable-length prosody and elaborated on this concept for fine-grained prosody control.

Wang et al. came up with the global style token (GST) Tacotron to encode different speaking styles [5]. Although they used the same reference encoder architecture used in earlier work [4], they did not use p itself for prosody embedding. Using a content-based attention, they computed the attention weights for style tokens from p . The attention weights represent the contribution of each style token, and the weighted sum of the style tokens is now used for style embedding. During the training step, each randomly initialized style token learns the speaking style in an unsupervised manner. In the inference mode, it was possible to control prosody by either predicting the style embedding from the reference speech or specifying the attention weights of the style tokens. This enables explicit control of the speaking style, but it nonetheless worked only in a global sense. If we are interested in controlling the prosody of a phoneme, it would be ideal to obtain the same prosody for different phonemes when the phonemes are conditioned on the same prosody embedding. However, GST Tacotron generates various types of prosody

for input phonemes that are conditioned on the same style embedding, which is not desirable for prosody control. Wang et al. also proposed text-side style control using multiple style embeddings for different segments of input text. This method could roughly change the style of the text segments, but it is limited when used to control phoneme-wise prosody for the reasons mentioned above.

3. BASELINE MODEL

We used a simplified version [10] of Tacotron for the base encoder-decoder architecture, but we used the original Tacotron [1] style of the Post-processing net and the Griffin-Lim algorithm [11] for spectrogram-to-waveform conversion. For the encoder input x , we used the phoneme sequence of normalized text to ease the learning. The one-hot speaker identity is converted into speaker embedding vector s by the embedding lookup layer. Equation 1 describes the base encoder-decoder, where e , p , and d denote the text encoder state, variable-length prosody embedding, and decoder state, respectively.

$$\begin{aligned} e_{1:l_e} &= \text{Encoder}(x_{1:l_e}) \\ \alpha_i &= \text{Attention}(e_{1:l_e}, d_{i-1}) \\ e'_i &= \sum_j \alpha_{ij} e_j \\ d_i &= \text{Decoder}(e'_i, s) \end{aligned} \quad (1)$$

Reference speech is encoded to prosody embedding using the reference encoder [4]. A mel-spectrogram of the reference speech proceeds through 2D-convolutional layers. The output of the last convolutional layer is fed to a uni-directional GRU. The last output of GRU r_N is the fixed-length prosody embedding p . If we use every output of GRU $r_{1:N}$ for prosody embedding, it forms the variable-length prosody embedding $p_{1:N}$.

4. PROPOSED METHOD

Fine-grained prosody control can be done by adjusting the values of variable-length prosody embedding. We propose two types of prosody control methods: speech-side control and text-side control. Variable-length prosody embedding is used as a conditional input at the encoder module or at the decoder module for speech-side control or text-side control, respectively. In order to do this, we need to align and downsample the prosody embedding to match the length of the prosody embedding l_p with the speech side (the number of decoder time-steps, l_d) or the text side (the number of encoder time-steps, l_e).

4.1. Modifications in the reference encoder

We empirically found that the following modifications improved the generation quality. We used CoordConv [12] for the first convolutional layer. According to its construction, Coordconv can utilize positional information while losing the translation invariance. We speculate that the positional information was helpful to encode prosody sequentially. We used ReLU as the activation function to force the values of the prosody embedding to lie in $[0, \infty]$.

The proposed models are trained identically to the Tacotron model. The model is trained according to the L1 loss between the target spectrogram and the generated spectrogram, and no other supervision is given for the reference encoder. Unless otherwise stated, we used the same hyperparameter settings used in earlier work [4].

4.2. Speech-side prosody control

The length l_p of variable-length prosody embedding created from a reference spectrogram with length l_{ref} is identical to l_{ref} . Note that the decoder should generate the same spectrogram as the reference spectrogram and that r -frames are generated at each decoder time-step. This gives l_p a longer length by r -times than l_d . By choosing appropriate stride sizes for the convolutional layers, we could shorten reference spectrogram to match l_p with l_d .

At each decoder time-step i , p_i is initially fed to the attention module together with $e_{1:l_e}$ to compute the i -th attention weights, α_i . We did not feed speaker embedding to the attention module as we assumed the speaker identity to be conditionally independent with attention weights when prosody is given. The weighted sum of $e_{1:l_e}$ with α_i gives us the context vector e'_i . The input of the decoder module at the i -th time-step is a concatenation of $\{e'_i, p_i, s\}$.

$$\begin{aligned} e_{1:l_e} &= \text{Encoder}(x_{1:l_e}) \\ \alpha_i &= \text{Attention}(e_{1:l_e}, p_i, d_{i-1}) \\ e'_i &= \sum_j \alpha_{ij} e_j \\ d_i &= \text{Decoder}(e'_i, p_i, s) \end{aligned} \quad (2)$$

4.3. Text-side prosody control

The linear relationship between l_p and l_d made it easy to ensure that speech-side prosody embedding has a length identical to the number of the decoder time-steps. Unfortunately, such a relationship is not guaranteed between l_p and l_e . We introduced a reference attention module that uses scaled dot-product attention [13] to find the alignment between $e_{1:l_e}$ and $p_{1:l_{ref}}$. In the reference attention module, *key* κ and *value* v are obtained from p and the *query* is e . Conceptually, the attention mechanism computes the attention weight according to the similarity between the *query* and each *key*, and weighted sum of the *values* is then obtained using the attention weight. To obtain κ and v from prosody embedding, we doubled the output dimension h of the reference encoder for the text-side prosody control, with the output split into two matrices of size $(l_{ref} \times h)$. The weighted sum of $v_{1:l_{ref}}$ with β gives us text-side prosody embedding p^t . Then, p^t is concatenated to e upon every usage of e .

$$\begin{aligned} e_{1:l_e} &= \text{Encoder}(x_{1:l_e}) \\ [\kappa_{1:l_{ref}}; v_{1:l_{ref}}] &= p_{1:l_{ref}} \\ \beta_j &= \text{Ref-Attention}(e_j, \kappa_{1:l_{ref}}) \\ p_j^t &= \sum_k \beta_{jk} v_k \\ \alpha_i &= \text{Attention}([e_{1:l_e}; p_{1:l_e}^t], d_{i-1}) \\ e'_i &= \sum_j \alpha_{ij} [e_j; p_j^t] \\ d_i &= \text{Decoder}(e'_i, s) \end{aligned} \quad (3)$$

4.4. Prosody normalization

Prosody embedding is normalized using each speaker's prosody mean. During training, we computed the sample mean along the temporal dimension of variable-length prosody embedding and stored average of the sample mean for each speaker. For both the training step and the evaluation, normalization was done by subtracting the speaker-wise prosody mean from every time step of prosody embedding.

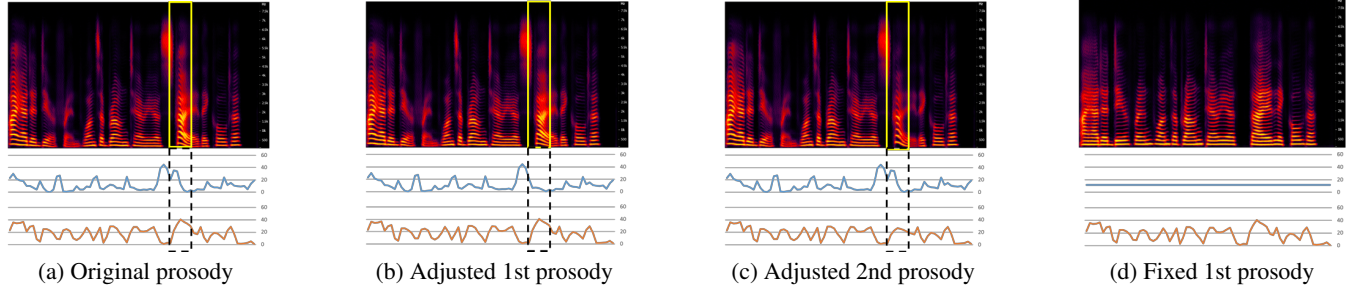


Fig. 1: Speech-side prosody control.

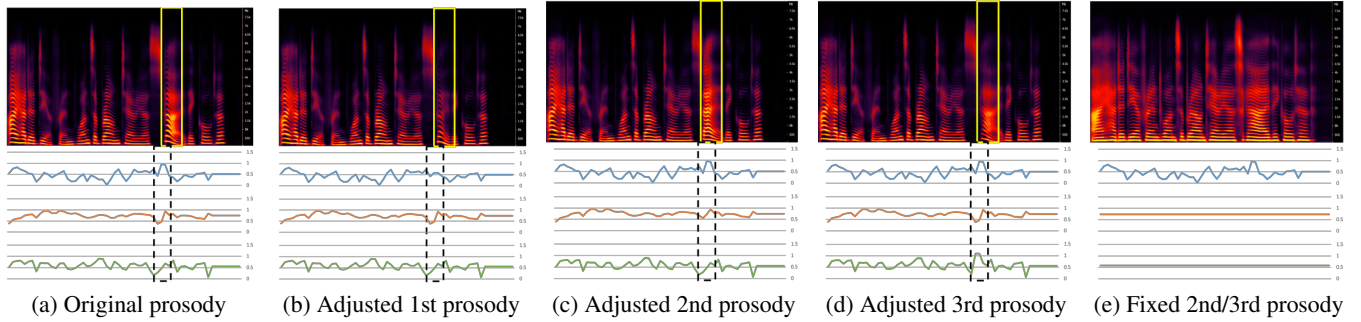


Fig. 2: Text-side prosody control.

5. EXPERIMENTS AND RESULTS

5.1. Dataset

Previous works [4, 5] used large amounts of data to train the prosodic TTS model (296 hours of data for the multi-speaker model). To ensure a large amount of data, we used multiple datasets, in this case VCTK, CMU ARCTIC, and internal datasets. The final dataset consisted of 104 hours (58 hours of English and 46 hours of Korean) with 136 speakers (128 English speakers and 8 Korean speakers).

Because variable-length prosody embedding has a large enough capacity to copy the reference audio, we had to use a very small dimension for the bottleneck size. This led us to the use of prosody sizes of 2 and 4 for the speech-side and text-side prosody embedding, respectively.

5.2. Speech-side control of prosody

By adjusting the values of the speech-side prosody embedding, we could change the prosody at a specific frame. Figure 1 shows the change in the learned prosody embeddings (line graph) and their corresponding spectrograms. The first dimension of prosody embedding, in the second row of Figure 1, tended to control the pitch of the generated speech. By comparing the highlighted parts of Figures 1-(a) and (b), one can assess the change of the pitch from the spaces between the harmonics. The second dimension of prosody embedding, in the third row of Figure 1, tended to control the amplitude of the generated speech. By comparing the highlighted parts of Figures 1-(a) and (c), one can assess the change of the amplitude from the intensity of the harmonics. We recommend that readers listen to the examples on our demo page.¹

¹<http://neosapience.com/research/2018/10/29/icassp>

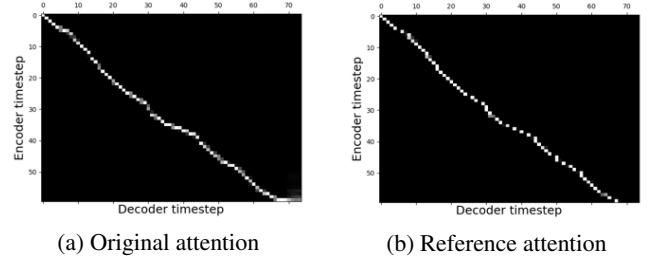


Fig. 3: Attention alignment between text and speech

5.3. Text-side control of prosody

First, we checked if the reference attention module learned how to find the alignment between the phoneme sequence and the reference audio. Figure 3 shows an attention alignment plot of the original attention module (a) and reference attention module (b). From their analogous shape, we find that the reference attention module could align the reference speech to the text.

As was done in Section 5.2, we changed the prosody of the phonemes by adjusting the text-side prosody embedding in Figure 2. It appeared that the amplitude was affected by the first and third dimensions and that the pitch was affected by the second and third dimensions. In addition, the length was affected by the first and third dimensions. It would be ideal if each dimension represents one prosodic feature (i.e., the pitch, amplitude, or length). We think prosody embedding is entangled because we did not impose any constraints on prosody embedding to be disentangled.

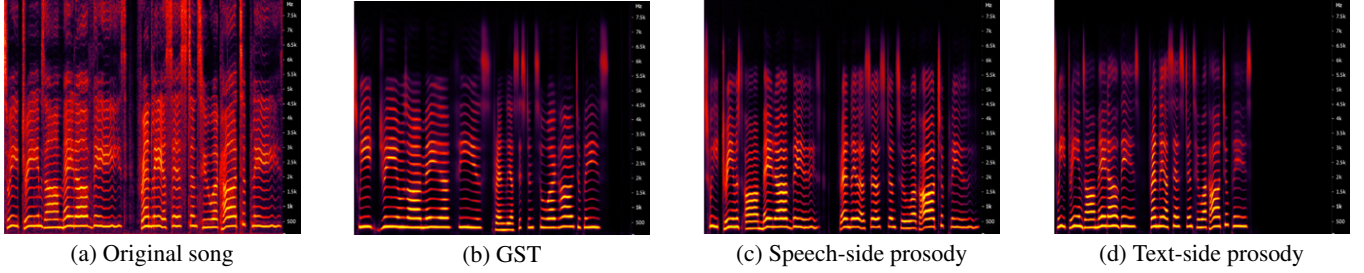


Fig. 4: Spectrogram from singing voice transfer.

5.4. Comparison with GST Tacotron

We compared our methods to GST Tacotron both quantitatively and qualitatively. For the quantitative comparison, we used the Mean Cepstral Distortion (MCD) with the first 13 MFCCs, as proposed in earlier work [4]. Table 1 shows that the proposed methods outperform GST Tacotron in terms of MCD_{13} , where a lower MCD is better. In particular, speech-side prosody control, which has the highest temporal resolution of prosody embedding, showed the lowest MCD.

Table 1: Mean cepstral distortion of types of prosody embedding

Model	MCD_{13}
Global style token	0.413
Speech-side prosody control	0.294
Text-side prosody control	0.342

One shortcoming of GST Tacotron is that GST works only in a global sense. If we fix GST for multiple decoder time steps, the decoder generates speech while changing the prosody implicitly at each time step to create the GST’s speech style. This is not problematic if the generated prosody perfectly matches the intention of the user, but in many cases we needed modifications to realize this. Because GST changes the prosody implicitly, it is ambiguous to control the prosody at specific moment. On the other hand, the proposed prosody embeddings control the prosody explicitly. In Sections 5.2 and 5.3, we observed that prosody can be controlled by adjusting the values of the prosody embedding. We further demonstrated the explicitness and consistency of the proposed methods by fixing prosody embedding to have the same value over all frames. This should give us a flat speech style in contrast to the GST approach, and we can see these outcomes in Figure 1-(d) and Figure 2-(e). The results are obtained by fixing the dimensions that controlled the pitch.

If we apply prosody control to a wider dynamic range of prosody embedding, it will be able to generate a singing voice. We demonstrate this in Figure 4. Using the three prosody control methods, we extracted the prosody from an unseen song of an unseen singer. We combined the extracted prosody embedding with the original lyrics and a speaker in the training set to perform the prosody transfer. While GST could not reconstruct the melody of the song, we could recognize the melody of the original song using the proposed methods. In particular, the generated song from speech-side prosody control was almost identical to the original song. For this task, the lyrics, speaker identity, and prosody embedding were the only requirements for the generation step. We witnessed the capability of the proposed methods to generate a song given an appropriate sequence of prosody

embedding.

5.5. Inter-speaker prosody transfer

We compared the MCD of the speech-side prosody embeddings with and without normalization, as shown in Table 2. For each case of prosody embedding, we computed the MCD between the reference and the generated speech for each prosody reconstruction and prosody transfer task. In both tasks, we used the speech of a female speaker as the reference speech, and we used the speaker ID of the same female speaker or another male speaker for prosody reconstruction and prosody transfer, respectively. Without normalization, the generated speech tended to show a higher pitch than the male speaker and sometimes failed to generate speech. We consider that this failure arises because the combination of the male speaker ID and female prosody embedding did not exist during the training step. When we used normalization for prosody embedding, the model was exposed to the similarly distributed prosody embedding during the training phase. This caused the prosody transfer to be easier compared to that without normalization. Table 2 also presents this phenomenon with a higher MCD during the prosody transfer with the non-normalized model compared to that with the normalized model.

Table 2: Mean cepstral distortion of types of prosody transfer

Model	Female-to-Female	Female-to-Male
Normalized	0.329	0.518
Not-normalized	0.304	0.531

6. CONCLUSION AND FUTURE WORK

Here, we proposed temporally structured prosody embedding networks to control the expressive style of synthesized speech. The proposed methods changed the pitch and amplitude both at the frame-level and phoneme-level resolution. Moreover, normalized prosody embedding made the prosody transfer step more robust to pitch discrepancies between the reference and generated speaker. The proposed methods demonstrated better quality in terms of the MCD score, and the prosody of a song could be successfully transferred to another speaker, resulting in voice conversion of a song.

The bottleneck size was the only factor that regularized the prosody embedding network in this paper. Disentangling techniques will be beneficial to factorize the prosody embeddings into more explainable prosodic features and separate them from other speech features. This will be a fruitful direction for future work.

7. REFERENCES

- [1] Yuxuan Wang, R.J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech 2017*, 2017, pp. 4006–4010.
- [2] Andrew Gibiansky, Serkan Arik, Gregory Diamos, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 2962–2970.
- [3] Eliya Nachmani, Adam Polyak, Yaniv Taigman, and Lior Wolf, "Fitting new speakers based on a short untranscribed sample," in *Proceedings of the 35th International Conference on Machine Learning*. 2018, pp. 3683–3691, PMLR.
- [4] RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," in *Proceedings of the 35th International Conference on Machine Learning*. PMLR, 2018, vol. 80, pp. 4693–4702.
- [5] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proceedings of the 35th International Conference on Machine Learning*. PMLR, 2018, vol. 80, pp. 5180–5189.
- [6] Daisy Stanton, Yuxuan Wang, and RJ Skerry-Ryan, "Predicting expressive speaking style from text in end-to-end speech synthesis," *arXiv preprint arXiv:1808.01410*, 2018.
- [7] Kim E. A. Silverman, Mary E. Beckman, John F. Pitrelli, Mari Ostendorf, Colin W. Wightman, Patti Price, Janet B. Pierrehumbert, and Julia Hirschberg, "Tobi: a standard for labeling english prosody," in *International Conference on Spoken Language Processing*. 1992, ISCA.
- [8] Colin W Wightman, "Tobi or not tobi?," in *Speech Prosody 2002, International Conference*, 2002.
- [9] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. 2014, pp. 1724–1734, Association for Computational Linguistics.
- [10] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ-Skerry Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu, "Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, 2018, pp. 4779–4783.
- [11] Daniel W. Griffin and Jae S. Lim, "Signal estimation from modified short-time fourier transform," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '83, Boston, Massachusetts, USA, April 14-16, 1983*, 1983, pp. 804–807.
- [12] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski, "An intriguing failing of convolutional neural networks and the convconv solution," *arXiv preprint arXiv:1807.03247*, 2018.
- [13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017.