KOREAN SINGING VOICE SYNTHESIS BASED ON AUTO-REGRESSIVE BOUNDARY EQUILIBRIUM GAN

Soonbeom Choi, Wonil Kim, Saebyul Park, Sangeon Yong, Juhan Nam

Graduate School of Culture Technology, KAIST, Daejeon, South Korea

ABSTRACT

Singing voice synthesis is a generative task that involves not only multidimensional controls of a singer model such as phonetic modulation by lyrics and pitch control by music score but also expressive elements such as breath sounds and vibrato. Recently, end-to-end learning models based on generative adversarial network (GAN) have drawn much interest as it requires less domain-specific processing but provides high sound quality. When GAN is applied to the audio domain, it entails several issues: the choice of audio representation to generate, handling temporal continuity between two adjacent outputs, and finding an effective loss metric for the audio representation. In this paper, we propose a Korean singing voice synthesis system that addresses the issues using an auto-regressive algorithm that generates spectrogram with the boundary equilibrium GAN objective. Through the qualitative test, we show the proposed methods are superior to the original GAN objective and non-auto-regressive model. We also show that our proposed method can render natural expressions such as continuous pitch contours and breath sounds.

***Index Terms*—** singing voice synthesis, auto-regressive model, boundary equilibrium GAN

1. INTRODUCTION

Singing Voice Synthesis (SVS) is a generative task that produces acoustic waveforms of singing given lyrics and melody input. Many of state-of-the-art SVS systems are based on concatenative synthesis or statistical parametric methods such as hidden Markov models. Concatenative synthesis can provide high-quality sounds but it requires a large amount of audio samples and analysis for seamless rendering [1, 2]. Statistical parametric methods allow for more compact implementation but they have multiple pipelines of processing modules [3]. Recently, end-to-end learning models based on deep neural networks have achieved remarkable results in speech synthesis [4, 5, 6], and this in turn has affected the development of neural singing synthesis [7]. The deep neural network method has advanced supporting diverse languages such as English [7, 8], Korean [9, 10], and Japanese [11, 12].

Unlike speech synthesis, the duration of pitched sounds is given from the music score in SVS. For example, score representations such as piano roll provide a temporal guide to render corresponding pitch and acoustics features. As a result, the score and rendered audio are well aligned to each other. Leveraging this metrical synchronization, researchers have attempted to generate vocoder parameters aligned with the score input using conditional generative adversarial neural network (GAN) [8, 12]. In this work, we recast the problem as an image-to-image translation where the input is a 2D time-pitch representation (e.g, piano roll) and the output is a 2D time-frequency representation [13]. That is, we use conditional GAN to generate spectrogram rather than vocoder parameter.

A fundamental issue in the image-based approach when applied to the audio domain is that the model can span only a short audio segment (e.g., several hundreds of audio frames) and therefore successively generated segments can be discontinuous over time. To address this problem, we propose an auto-regressive conditional GAN which uses spectrogram in a previous time step as input to produce spectrogram in the current time step. In addition, we apply boundary equilibrium GAN (BEGAN) to achieve stable training of the model and enhance the quality of generated output [14]. Through listening test, we show that the proposed method is generally superior to the original GAN and non-auto-regressive settings.

2. RELATED WORK

The proposed system is based on modules from speech synthesis and image generation. Recently, end-to-end speech synthesis based on auto-regressive methods such as Tacotron and Deep Voice outperformed concatenative synthesis [4, 5, 15]. Many of SVS systems followed the advance of speech synthesis systems because both systems share the text-to-voice transformation [8, 9, 10, 11, 12]. In image generation, many systems have been proposed for high quality image generation and, among others, GANs have shown superior performances [16, 17]. Several SVS systems adopted GANs to generate high-quality acoustic features [8, 10, 12].

An important issue in applying GANs to the audio domain is the choice of acoustic representations to generate the final waveforms. Hono et al. and Chandra et al. used off-the-self vocoders to obtain reliable sound quality [8, 12]. Lee et al. generated mel-spectrogram and then transform it to linear

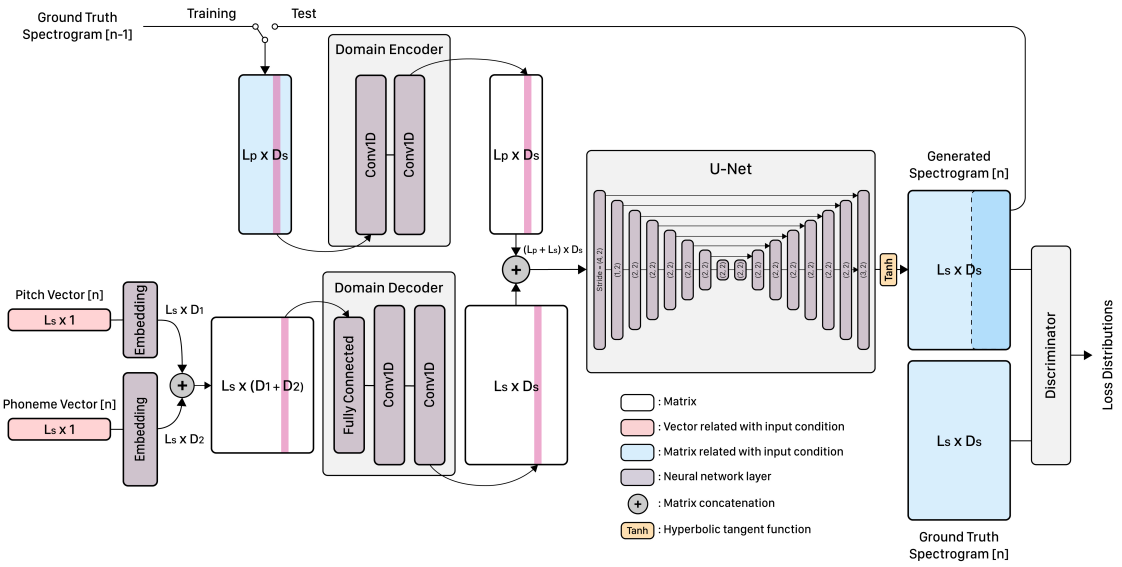

Fig. 1. Overview of the proposed singing voice synthesis system. L_s and L_p are the length of the generated spectrogram and the previous spectrogram, respectively. D_1 and D_2 are embedding dimensions for pitch and phoneme, respectively. D_s is the number of frequency bins of the spectrograms.

spectrogram using a super-resolution network in a more end-to-end fashion [10]. In this paper, we directly generate linear spectrograms and show the possibility. Another issue in training GANs is the choice of loss metrics, which affects the stability in balancing between the generator and the discriminator and in turn better quality of the generated output. In this paper, we adopt BEGAN loss [14] and show that it outperforms the original GAN loss when they are used to generate linear spectrogram.

3. PROPOSED SYSTEM

3.1. Overall Processing

Figure 1 illustrates the overview of the proposed system. It generates spectrogram with L_s frames and D_s frequency bins from a pitch vector and a phoneme vector with L_s frames and the previous spectrogram with L_p frames and D_s frequency bins. The input vectors and the spectrograms have the same frame rate. The pitch vector and the phoneme vector are separately embedded and concatenated as a single matrix with the size of $L_s \times (D_1 + D_2)$. We applied the domain encoder and decoder to generate hidden-layer outputs with the size of $L_s \times D_s$ and $L_p \times D_s$ from the pitch/phoneme vector and from the previous spectrogram. Two hidden-layer outputs are concatenated and followed by a U-Net to generate spectrogram [18]. In training phase, the previous spectrogram is taken from the ground truth spectrogram before L_p frame but, in test phase, it is from the last L_p frame of the previously generated spectrogram. Finally, we used the Griffin-Lim algorithm to reconstruct phase information and generate waveforms [19]. The discriminator that takes the generated spectrogram and the ground-truth spectrogram is trained using pixel-wise L1 loss and the BEGAN loss.

3.2. Input Processing

Our system uses score and text information to generate singing voice. A single note in a score contains pitch and duration information and it is converted into a vector. The duration from the vector is converted to have the same length as spectrogram in time. The pitch range is determined by the minimum and maximum pitch values in the dataset. We extracted phonemes from text information using Korean grapheme-to-phoneme algorithm [20]. Korean phonemes can be divided into onset, nucleus and coda and we located onset and coda at the first and the last frames of a note [10]. Nucleus represents voiced sound and thus it is elongated to be proportion to the length of note.

We sampled audio in 22050Hz and applied pre-emphasis with 0.97 filter coefficient to it. We computed spectrograms using 1024-point fast Fourier transform and compressed them in dB and normalized the scale to use a hyperbolic tangent nonlinearity. We scaled the range to [-0.8, 0.8] instead of [-1.0, 1.0] because we found that it helps allowing outliers and generating envelope closer to the ground truth [21].

3.3. Domain Encoder/Decoder

We use score vectors, text vectors and previous audio spectrogram as input to generate current spectrogram. Concatenating the different domains of data is not appropriate to use convolution kernels in the following U-Net. Therefore, we add a domain encoder and decoder to convert the different domains of data into a unified representation with a similar level of abstraction. Figure 2 shows how the domain encoder and decoder are converted to a hidden-layer representation and the concatenation are to generate the spectrogram via the U-Net. The domain decoder consists of one fully connected layer and

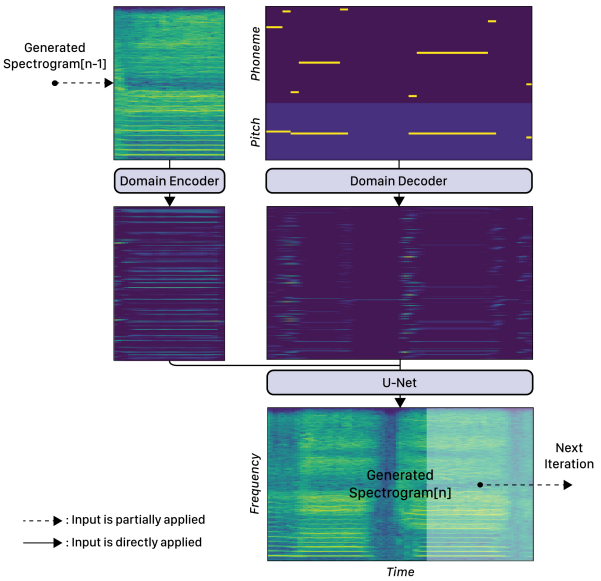

Fig. 2. The workflow of the domain encoder and decoder.

two 1D convolutional layers with ReLU activation function. The domain encoder consists of two 1D convolutional layers with ReLU activation function. They process inputs frame-wise and both structures are inspired from [15].

3.4. U-Net

U-Net has recently shown superior performance as a spectrogram generator in singing voice separation [22, 23]. In our system, the U-Net converts the outputs of the domain encoder/decoder to spectrogram. The encoder of the U-Net has striding convolution kernels instead of max-pooling referring to deep convolutional GAN (DCGAN) [17]. Each convolution layer in the encoder is down-sampled to the half in size except for the first layer and every layers use a LeakyReLU activation function. In the decoder of the U-Net, each convolution layer is up-sampled to the double in size using transposed convolution except for the last layer and every layer use an ReLU activation function. The first layer of the U-Net down-samples its input by 4×2 and the last layer up-samples its input by 3×2 so that the U-Net generates current spectrogram from the previous spectrogram in time.

3.5. Discriminator

BEGAN uses an auto-encoder for discriminator [14]. While the original GAN matches the distributions between real and generated samples directly, BEGAN balances discriminator and generator using the auto-encoder loss. This allows more stable training. More specifically, BEGAN relaxes the equilibrium of the auto-encoder loss using a hyper-parameter $\gamma \in [0, 1]$ defined as:

$$\gamma = \frac{\mathbb{E}[\mathcal{L}(G(x))]}{\mathbb{E}[\mathcal{L}(y)]} \tag{1}$$

where y is a real sample, $G(x)$ is a generated sample, and $\mathcal{L}$ is the reconstruction error of the auto-encoder. Lower values of γ put more weight on auto-encoding real samples. Thus it provides higher quality but lower diversity in generated samples. On the other hand, higher values of γ provides higher diversity but lower quality. The loss functions of discriminator and generator are defined as:

$$\begin{cases} \mathcal{L}_D = \mathcal{L}(y) - k_t \mathcal{L}(G(x)) \\ \mathcal{L}_G = \mathcal{L}(G(x)) + |y - G(x)| \\ k_{t+1} = k_t + \lambda(\gamma \mathcal{L}(y) - \mathcal{L}(G(x))) \end{cases} \tag{2}$$

where λ works as a learning rate and k_t is an updating parameter that maintains equation 1. In our setting, x is a sample from the domain encoder/decoder output and y is a sample from ground truth spectrogram. Note that we add the L1 loss to the generator loss ($\mathcal{L}_G$) to reflect the pixel distribution between the generated spectrogram and the ground truth. We used stacks of convolutional layers and exponential linear units in the auto-encoder, following the configuration in [14].

4. EXPERIMENTS

4.1. Dataset

We collected our own dataset due to the absence of open datasets for Korean singing voice synthesis. The dataset is composed of 50 Korean children songs and they are recorded by one professional female singer. Each song is recorded in two separate keys and the difference between keys are range from 3 to 5 semitones. The total duration of the recordings is about 2 hours and 38 minutes. We also collected MIDI files that contain melodic notes and text files that contains lyrics. For each song, we manually aligned audio recordings to corresponding notes and syllables. We split the dataset into 41 songs for training, 1 song for validation and 8 songs for test. We used the single song in the validation set to monitor the loss and check the generation quality by listening while training the model.

4.2. Experiment Settings

We used the original GAN objective without auto-regressive method as a baseline model. Its discriminator is based on DCGAN [17] and it predicts the real and generated spectrogram. Given the baseline model, we changed the following factors:

- Application of the BEGAN objective in Section 3.5.
- Changes of γ in the BEGAN objective.
- Auto-regressive module in Section 3.3.

Table 1 compares the different settings. In the five models, the network configurations are fixed except the settings. We used the Adam optimizer [24] for both the generator and the discriminator with $\beta 1 = 0.5$ and $\beta 2 = 0.999$. The learning rate starts from 10^{-4} and it is reduced by half in every 50 epochs.

Table 1. Compared models and the experiment settings.

Model	Model 1	Model 2	Model 3	Model 4	Model 5
Type of GAN	Original GAN	Original GAN	BEGAN	BEGAN	BEGAN
γ in BEGAN	-	-	1.0	1.0	0.7
Auto-regressive	No	Yes	No	Yes	Yes

Table 2. Qualitative evaluation results in MOS.

Model	Pronunciation Acc.	Sound Quality	Naturalness
Model 1	2.267 ± 0.988	1.991 ± 0.826	2.099 ± 1.021
Model 2	2.052 ± 0.993	1.896 ± 0.876	2.099 ± 1.095
Model 3	$\mathbf{3.070 \pm 1.003}$	2.788 ± 0.924	2.867 ± 1.045
Model 4	3.038 ± 1.057	$\mathbf{2.965 \pm 0.955}$	$\mathbf{3.122 \pm 1.074}$
Model 5	2.646 ± 1.021	2.377 ± 0.904	2.519 ± 0.997
Reconstruction	4.681 ± 0.645	4.333 ± 0.713	4.600 ± 0.717
Ground Truth	4.780 ± 0.564	4.701 ± 0.656	4.762 ± 0.582

4.3. Evaluation

4.3.1. Qualitative evaluation

We evaluated the results by a listening test. We segmented the test set into 15 examples by removing silence and repeating parts. We included not only the generated samples from the five models but also a reconstructed sample and the ground truth sample. The reconstructed sample was obtained directly from the ground truth spectrogram via the Griffin-Lim algorithm. 23 participants evaluated the results with Mean Opinion Score (MOS) in three criteria (pronunciation accuracy, sound quality, and naturalness) [10]. The participants are graduate students working on speech or singing voice research.

The MOS results are shown in Table 2. Compared to the original GAN models (model 1 and 2), BEGAN models (model 3 and 4) have significantly higher scores in all of three criteria. Between the two BEGAN models, the auto-regressive method (model 4) has higher scores than the non-auto-regressive method (model 3) except for pronunciation accuracy. To investigate the results further, we conducted a paired t-test between model 3 and 4. The obtained p-values for pronunciation accuracy, sound quality and naturalness are 0.5707, 0.0007, and 0.0001, respectively. This indicates that the difference in pronunciation accuracy between two models are statistically insignificant whereas those in sound quality and naturalness are statistically significant. Therefore, we can conclude that the auto-regressive method generally helps improving the quality of singing voice synthesis. Between the two original GAN models, however, the auto-regressive one (model 2) shows worse performance than the non-auto-regressive one (model 1). This might be because the auto-regressive model allows the low quality of generated spectrogram from the previous time to adversely affect the generation in the current time. The result for model 5 shows that the lower γ value leads to lower quality unlike in [14]. This might be because our setting includes the additional L1 loss term and it gains relatively more weight than $\mathcal{L}(G(x))$ for lower γ in the generator loss.

While the proposed method (model 4) improves the overall performance, it still has limitations, for example, chorus effect in sound quality, incorrect pronunciation and other artifacts[1]. We suspect that the chorus effect is partly from the the Griffin-Lim algorithm, and incorrect pronunciation and other artifacts are from unseen pairs of a note and phonemes in the test set, in other words, the training set is not sufficient to cover all possible combinations of notes and phonemes.

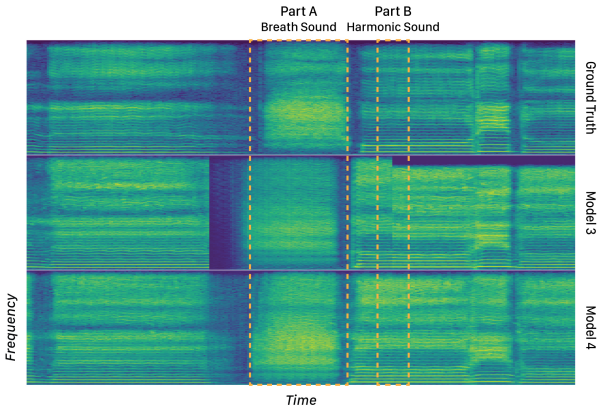

Fig. 3. The ground truth and generated spectrograms of model 3 and model 4.

4.3.2. Spectrogram analysis

We inspect the generated spectrogram further to validate the proposed method. Part A in Figure 3 compares the breath sound of the ground truth and the generated spectrograms from model 3 and 4. It shows that the both models can generate the breath sound which may come from the routine of the singer. This is possible only when they learn temporal context well in the audio recording. Between the two models, the auto-regressive one (model 4) has more natural formant shapes. In addition, Part B shows that the auto-regressive method enables the model to generate continuous spectrogram without abrupt changes in harmonic tone generation.

5. CONCLUSION

We proposed Korean singing voice synthesis based on auto-regressive boundary equilibrium GAN. We showed the proposed methods are superior to the original GAN and non-auto-regressive model. For future work, we plan to increase the volume of the dataset for more reliable training of the model and also find more compact audio representations to improve the sound quality.

6. ACKNOWLEDGEMENT

This material is based upon work supported by the Ministry of Trade, Industry & Energy (MOTIE, Korea) under Industrial Technology Innovation Program (No. 10080667, Development of conversational speech synthesis technology to express emotion and personality of robots through sound source diversification).

[1]The audio examples of the best model are available at: `https://soonbeomchoi.github.io/saebyulgan-blog/`

7. REFERENCES

[1] J. Bonada, A. Loscos, and H. Kenmochi, "Sample-based singing voice synthesizer by spectral concatenation," in *Proceedings of Stockholm Music Acoustics Conference*, 2003, pp. 1–4.

[2] H. Kenmochi and H. Ohshita, "Vocaloid - commercial singing synthesizer based on sample concatenation," in *Proc. INTERSPEECH*, 2007, pp. 4009–4010.

[3] K. Oura, A. Mase, T. Yamada, S. Muto, Y. Nankaku, and K. Tokuda, "Recent development of the HMM-based singing voice synthesis system—Sinsy," in *Seventh ISCA Workshop on Speech Synthesis*, 2010, pp. 803–806.

[4] Y. Wang, R.J. Skerry-Ryan, D. Stanton, Y. Wu, R.J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, and Q. Le, "Tacotron: Towards end-To-end speech synthesis," in *INTERSPEECH*, 2017.

[5] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 2962–2970.

[6] A.V.D. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[7] M. Blaauw and J. Bonada, "A neural parametric singing synthesizer modeling timbre and expression from natural songs," in *INTERSPEECH*, 2017, pp. 4001–4005.

[8] P. Chandna, M. Blaauw, J. Bonada, and E. Gomez, "Wgansing: A multi-voice singing voice synthesizer based on the wasserstein-gan," *arXiv preprint arXiv:1903.10729*, March 2019.

[9] J. Kim, H. Choi, J. Park, M. Hahn, S. Kim, and J. Kim, "Korean singing voice synthesis system based on an LSTM recurrent neural network," in *INTERSPEECH*, September 2018, pp. 1551–1555.

[10] J. Lee, H. Choi, C. Jeon, J. Koo, and K. Lee, "Adversarially trained end-to-end korean singing voice synthesis system," in *INTERSPEECH*, 2019, pp. 803–806.

[11] M. Nishimura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Singing voice synthesis based on deep neural networks," in *INTERSPEECH*, 2016, pp. 2478–2482.

[12] Y. Hono, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Singing Voice Synthesis Based on Generative Adversarial Networks," in *ICASSP*, 2019, pp. 6955–6959.

[13] P. Isola, J.Y. Zhu, T. Zhou, and A.A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in *CVPR*, 2017, pp. 5967–5976.

[14] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: Boundary Equilibrium Generative Adversarial Networks," *arXiv preprint arXiv:1703.10717*, 2017.

[15] J. Shen, R. Pang, R.J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R.j. Skerrv-Ryan, R. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *ICASSP*, 2018, pp. 4779–4783.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[17] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[18] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Lecture Notes in Computer Science*, vol. 9351, pp. 234–241, October 2015.

[19] D.W. Griffin and J. Lim, "Signal Estimation from Modified Short-Time Fourier Transform," *ICASSP*, vol. 32, pp. 236–243, May 1984.

[20] Y. Cho, "Korean grapheme-to-phoneme analyzer (kog2p)," `https://github.com/scarletcho/KoG2P`, 2017.

[21] J. Engel, K.K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "Gansynth: Adversarial neural audio synthesis," *arXiv preprint arXiv:1902.08710*, 2019.

[22] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep u-net convolutional networks," in *ISMIR*, 2017, pp. 323–332.

[23] D. Stoller, S. Ewert, and S. Dixon, "Adversarial semi-supervised audio source separation applied to singing voice extraction," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2391–2395.

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