INTERACTIVE MULTI-LEVEL PROSODY CONTROL FOR EXPRESSIVE SPEECH SYNTHESIS

Tobias Cornille[⋆] *Fengna Wang*[†] *Jessa Bekker*[⋆]

[⋆] KU Leuven, Belgium
[†]Acapela-Group, Mons, Belgium

ABSTRACT

Recent neural-based text-to-speech (TTS) models are able to produce highly natural speech. To synthesize expressive speech, the prosody of the speech has to be modeled, and predicted/controlled during synthesis. However, intuitive control over prosody remains elusive. Some techniques only allow control over the global style of the speech and do not allow fine-grained adjustments. Other techniques create fine-grained prosody embeddings, but these are difficult to manipulate to obtain a desired speaking style. We thus present ConEx, a novel model for expressive speech synthesis, which can produce speech in a certain speaking style, while also allowing local adjustments to the prosody of the generated speech. The model builds upon the non-autoregressive architecture of FastSpeech and includes a reference encoder to learn global prosody embeddings, and a vector quantized variational autoencoder to create fine-grained prosody embeddings. To realize prosody manipulation, a new interactive method is proposed. Experiments on two datasets show that the model enables multi-level prosody control.

Index Terms— speech synthesis, text-to-speech, prosody, controllability, hierarchical prosody embedding

1. INTRODUCTION

Humans use expressive speech to convey more than words. We express emotions and even deliver additional meaning (e.g. irony) through our way of speaking. To do so, we vary speech characteristics such as intonation, stress, rhythm and so forth, collectively referred to as *prosody*. In order to generate realistic expressive speech, TTS models should thus produce the fitting prosody. Prosody modeling in TTS currently attracts great attention [1, 2, 3]. However, predicting prosody is hard and cannot entirely be solved, since the problem of mapping text to speech is underdetermined; one text can map to many plausible speech utterances with varied prosody. To alleviate this, control over the prosody of the synthesized can be useful, especially when the speech synthesis systems do not produce speech in the desired prosody.

Despite the tremendous progress made in expressive speech synthesis [4, 5, 6, 7, 8, 1, 9, 10, 2, 11], the existing literature is highly focused on *modeling* the prosody, but leaves an important gap where *controlling* the prosody in a practical way is considered. While controlling the global style can be done intuitively, by selecting the general style, or transferring it from a reference utterance, it is unclear how to control the prosody on a finer level. The research on fine-grained prosody in speech synthesis has been limited to modeling and predicting the prosody [1, 2, 3]. Just having the ability to model the fine-grained prosody is insufficient, because even if the representation is interpretable (e.g. pitch and duration), it is hard to set them all individually to obtain a desired and coherent speaking style.

This paper bridges the prosody controllability gap, by proposing **ConEx**: a novel interactively **Con**trollable model for **Ex**pressive speech synthesis. It enables control over the global speaking style of the generated speech, while still allowing local prosody adjustments in an intuitive way. The main insight is that, while it is arduous to describe prosody on a fine-grained level, it is very easy to assess through listening whether an utterance has the desired prosody. We propose an interactive method where the user indicates the phoneme from where onward the prosody needs to change. The model then generates new options with variations in the prosody, while maintaining coherency and the global style.

We make the following contributions: 1) ConEx, an extension of the FastSpeech TTS model [12], with a novel hierarchical prosody encoder that is the first to combine global speaking style and phoneme-level prosody. 2) A novel interactive method for making local edits to the prosody of synthesized speech. It allows locally changing the prosody, while still maintaining the desired global style and naturalness of the output speech. 3) Experiments which demonstrate that the proposed technique enables fine-grained prosody edits, such as emphasizing words or changing their duration, while maintaining coherency and the desired global style.

2. BACKGROUND

2.1. Text-To-Speech

The task of *speech synthesis* or *text-to-speech (TTS)* is to generate speech from some input text. Neural TTS models typ-

ically consist of two stages, where the first generates a mel-spectrogram from text [13, 12], and the second, the vocoder, synthesizes speech from the mel-spectrogram [14, 15, 16, 17, 18]. We focus on controlling the prosody in the first stage.

The first TTS stage is usually an encoder-decoder sequence-to-sequence model [19] that maps text to mel-spectrograms. Coherent speech can be obtained via an *autoregressive* decoder, as in Tacotron [13]. First, the *encoder* creates a contextualized representation of each input element. Then, the *decoder* transforms these representations into the output sequence. Other models, such as FastSpeech [12], Parallel Tacotron [2], Flow-TTS [20], and ParaNet [21] greatly improve the training and synthesis speed by using non-autoregressive architectures. To synthesize speech with coherent prosody, the prosody can either be learned from an autoregressive teacher model [12], or explicitly be modeled and predicted [22, 6, 8, 23, 9, 1, 2, 3].

2.2. Modeling Prosody

Prosody can be represented using interpretable prosodic feature, such as pitch, loudness, and duration [9, 24]. However, it is hard to set them so to create a desired prosodic effect. Alternatively, latent prosody representations can be learned. A prosody encoder network takes a speech fragment as its input and outputs an embedding which represents the prosody in that fragment [4, 6, 8, 2, 23].

The prosody can be modeled as representations of different granularity levels: 1) **Utterance-level** representations capture the global speaking style [4, 5, 6, 8, 2], 2) **Phoneme-level** embeddings represent fine-grained aspects (like duration, pitch and loudness) of a single phoneme [7, 9, 1, 2], 3) **Syllable-level** representations [10], and 4) **Hierarchical** approaches condition fine-grained representations (phonemes) on coarser representations (e.g. words) [23, 3].

This paper proposes a hierarchical approach with global (utterance-level) style, and phoneme-level prosody.

2.3. Controlling Prosody

Controlling prosody usually uses one of two general approaches: 1) Generate speech in a certain global speaking style. This style is obtained from a reference utterance, or a preset set of styles. In practice, variations in speech can be generated for the same style by "abusing" the model's stochasticity to generate multiple versions of the same speech. However, there is no way to steer this process. 2) Using fine–grained prosody features/embeddings, by setting them for each phoneme or syllable. Even with interpretable features, it is arduous to manipulate them to obtain the desired and coherent prosody. To the best of our knowledge, no intuitive method for controlling the fine-grained prosody has been proposed in the literature.

This paper extends global speaking style control with a new method for intuitively making local prosody adjustments.

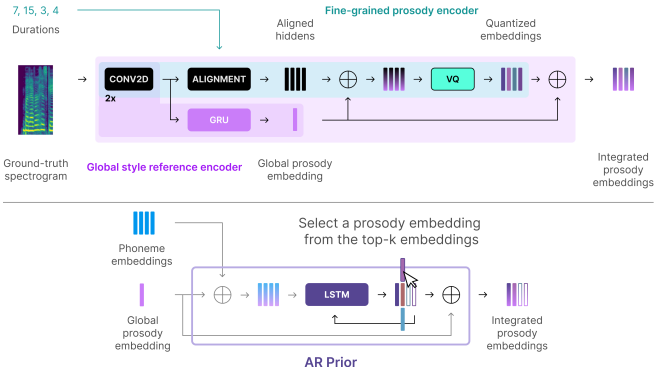

Fig. 1. The hierarchical prosody encoder. Top: the reference encoder and the fine-grained prosody encoder during training. Bottom: the AR prior at inference time.

3. CONEX: CONTROLLABLE EXPRESSIVE TTS

ConEx is a novel interactively **Con**trollable model for **Ex**pressive speech synthesis. To synthesize speech with the desired prosody, the user first provides text and a global style, and then interactively improves the prosody by giving local hints. The local hint is the phoneme until where the generated speech is satisfactory. ConEx then generates new options by changing the fine-grained prosody starting from the indicated phoneme, and the user selects the best option. This process is repeated to iteratively converge to the desired prosody.

To achieve this type of controllability, the prosody is represented hierarchically: at the global level (for the speaking style) and at phoneme level. The sequence of local prosody embeddings needs to be consistent with the global style as well as coherent utterance-wide. ConEx allows prosodic variations by supporting diverse fine-grained prosody embedding sequences per style.

To allow for diverse yet coherent prosody, ConEx employs vector-quantized fine-grained prosody embeddings in combination with an autoregressive prior over this embedding space, as this has shown promise [1]. We further extend it to condition the fine-grained prosody embeddings on the global style extracted from the reference utterance. Furthermore, we provide a method for interactive prosody control. Fast speech generation is essential to support real-time interaction, therefore, the architecture follows FastSpeech [12].

3.1. Architecture

The ConEx architecture follows FastSpeech [12] (with the duration predictor from FastSpeech 2 [9]), but adds a novel hierarchical prosody encoder. Figure 1 shows the three elements of this prosody encoder: 1) the reference encoder that extracts global style embeddings from reference utterances, 2) the VQ-VAE for quantized fine-grained prosody embeddings to locally control the prosody, and 3) the autoregres-

sive model that represents the conditional prior over the fine–grained prosody space.

1) The global style reference encoder encodes the speaking style of an entire utterance in a single embedding. The input is the mel-spectrogram of a reference utterance in the desired style, but possibly with different text. The output is a continuous global prosody embedding. The reference encoder, based on [4], has two 2D convolutional layers (to downsample the input), one GRU layer (to summarize over time), and a final projection.

2) The fine-grained prosody encoder encodes the phoneme-level prosody as discrete embeddings. Discrete embeddings are used for easier control, and because they can be predicted by a simple autoregressive prior model during synthesis [1]. The embeddings are encoded from their ground-truth mel-spectrogram using a *vector quantized variational autoencoder (VQ-VAE)* [25], adapted for hierarchical embeddings [3]. The encoder starts from the global reference encoder's downsampled mel spectrogram. The alignment layer averages the spectrogram frames per phoneme to align them with the phoneme embedding sequence. Two linear ReLU-activated projection layers project the phoneme vectors to a latent 3D space. Then vector quantization (VQ) is applied: each phoneme vector is replaced by the closest embedding from a codebook with k discrete embeddings. After a final projection, the fine-grained embeddings are added to the phoneme embeddings.

3) Fine-grained prosody prediction is necessary to synthesize speech for an unseen phoneme sequence, because the fine-grained prosody encoder cannot be used without a ground-truth mel spectrogram. An autoregressive (AR) prior model predicts the fine-grained prosody embeddings from the global style embedding and the phoneme sequence. The model is autoregressive to generate coherent prosody. Similar to [1], the AR prior model is an LSTM, but the input was extended by concatenating the style embedding to the phoneme embedding. The output is a categorical distributions over the different discrete embeddings.

3.2. Training

In a first step, everything except the AR prior is trained, using pairs of text (as phoneme sequences) and speech (as mel-spectrograms), so to minimize the MAE. The mel-spectrogram is the target as well as the prosody encoder input. A second step trains the AR prior model so to minimize the fine-grained prosody embeddings MSE, using the global style embeddings and phoneme embeddings as input, and quantized fine-grained prosody embeddings as targets, all generated by the model trained in the first step.

3.3. Interactively Controllable Speech Synthesis

The global speaking style is obtained by choosing a global prosody embedding from one of the training utterances or by using the reference encoder to encode a reference utterance with the desired style. Fine-grained prosody embeddings for this global style are generated by the AR prior model, as shown in figure 1. To make **local adjustments to the prosody** starting from a specific phoneme, the categorical distribution predicted by the AR prior model for that phoneme is used to obtain the top k options. The rest of the fine-grained prosody embedding sequence is generated autoregressively, resulting in k speech variations, which vary starting from the indicated phoneme. After selecting the preferred generated utterance, the user can employ the same method iteratively to make further adjustments, if desired. By employing the AR prior model which is conditional on the global style, the generated speech will sound natural as well as conform with the global style.

4. EXPERIMENTS

We aim to answer the following questions: (1) When no prosody adjustments are made, can ConEx generate speech with a desired global style? (2) Can fine-grained prosody embeddings be swapped to obtain the desired local prosody? The accompanying demo page[1] contains audio samples to support the experimental results.

4.1. Experimental Setup

Data Two datasets are used, both consisting of speech utterances annotated with their phoneme description and alignment. The first dataset, "Styles", is a proprietary dataset, consisting of 23.5 hours of speech in US English recorded by a professional voice actor. There are 13686 utterances, each annotated with one of six styles (normal, happy, sad, old, villain, loud). Between the samples of one style, the prosody does not vary a great deal. The second dataset: "blizzard", is the 2013 Blizzard Challenge dataset [26], consisting of 147 hours of speech from 49 audiobooks narrated by Catherine Byers. The utterances are highly expressive and contain ample prosodic variation. It has no additional annotations. Six representative styles were selected by choosing diverse embeddings from a 2D t-SNE plot.

Model & Training The proposed architecture was implemented as an extension to ESPnet [27]. ConEx was trained for 400.000 steps (42-48 hours on a single NVIDIA P100 GPU), using the Adam optimizer [28] (β_1=0.9, β_2=0.999, and ϵ=10^{-4}) and the learning rate scheduler from [29]. All hyperparameters are listed on the accompanying demo page. The AR prior model was trained with the same optimizer and schedule, but for 25.000 steps (around 3.5 hours) on the same hardware. After the training step, our speech synthesis method requires no GPU nor special powerful hardware to

[1] https://people.cs.kuleuven.be/~jessa.bekker/ConEx/

generate and control new speech samples. To allow interactive control, we used a fast vocoder: parallel WaveGAN [18], pre-trained on the LibriTTS corpus [30][2].

4.2. Global Style Transfer without Local Adjustments

To evaluate the global style transfer, an AXY test is carried out: for each input sentence two speech outputs, X and Y, are generated: X using a reference utterance A, Y without global style conditioning. If the style is transferred, then X should be more similar to A than Y. The experiment was carried out for all representative styles and three input sentences. The distance is measured by *Mel Cepstral Distortion (MCD)* [4] and the F_0 MSE [31]. The utterances are aligned using *Dynamic Time Warping (DTW)*. Additionally, a qualitative comparison can be done on the accompanying demo page.

Table 1 shows that the samples generated using the global prosody embedding generally closer to the reference sample. Qualitatively listening to the generated audio leads to the same conclusion for all styles of the "Styles" dataset, and styles 4-6 of the Blizzard dataset. However, styles 1-3 are transferred less clearly, as the speaking rate and rhythm do not seem to correspond.

Data	**Style**	**Mean MCD (↓)**		**Mean F_0 MSE (↓)**	
	AX	AY	AX	AY	
Styles	Happy	**3.61**	3.97	**6992.11**	7270.97
	Loud	**5.30**	6.61	**8102.28**	10089.48
	Old	**2.55**	3.05	**3721.09**	5635.80
	Sad	**4.16**	4.63	**3090.39**	5820.60
	Villain	**3.32**	3.83	**1164.74**	9991.15
Blizzard	Style 1	**3.49**	3.70	**11270.39**	11806.36
	Style 2	**3.44**	3.56	**11469.32**	11338.54
	Style 3	**3.29**	3.74	**14614.33**	14778.82
	Style 4	**3.22**	3.75	15728.65	**14835.17**
	Style 5	**3.56**	3.67	**9979.27**	10403.00
	Style 6	**3.79**	4.04	**15230.49**	15340.72

Table 1. Mean MCD and mean F_0 MSE values for the difference between the samples generated using a global prosody embedding and the reference sample (AX) and the difference between baseline samples and the reference sample (AY).

4.3. Fine-Grained Prosody Control

To evaluate whether fine-grained prosody embeddings can be swapped to obtain the desired local prosody, different qualitative tests are carried out, whereby the effects of changing the fine-grained prosody embeddings are analyzed. This approach is similar to the controllability experiments of [23, 10]. First, speech is generated using the global style. Then, the prosody embeddings for a certain phoneme were edited by selecting one of the other top-3 embedding options. Three text inputs were used to evaluate this method: 1) "I didn't say he stole the money" 2) "Whenever you feel like criticizing anyone, he told me, just remember that all the people in this world haven't had the advantages that you had" 3) "This would have changed the grand result of the war" For the first sentence, samples were generated to test if every word of the sentence could be stressed. The second and third sentence the "a" in "anyone" was adapated, and in the third "war".

For the Blizzard dataset, the method successfully emphasized 4 out of 7 words of sentence 1. The original prediction lead to an emphasis on "didn't". The phonemes corresponding to "I", "say", and "stole" could be emphasized by choosing different fine-grained prosody embeddings from the top 3, but "he", "the", and "money" could not. Changing the prosody embedding corresponding to "he" lead to a change in prosody for "say". The effects of the edit were thus not local. Furthermore, the "o" in "money" could only be emphasized by using two prosody embeddings that were not suggested by the AR prior. The majority of the other codes caused changes in the prosody of "stole". Sentence 2 show that the diversity of the the top-3 fine-grained prosody embeddings can sometimes be limited. When the prosody embedding for the phoneme corresponding to the "a" in "anyone" was changed for the second option, the output speech only changed slightly. Furthermore, some prosody embedding caused a change in a previous phoneme, again indicating that the effects are not entirely local. Sentence 3 also showed this effect. When the fine-grained prosody embedding of "wa" in "war" was swapped, the prosody of "sult" in "result" changed. These non-local effects complicate the process of making targeted edits to local prosody. The self-attention mechanism in the ConEx decoder is probably the reason of this problem, as it could allow a fine-grained prosody embedding to influence the prosody of more phonemes than only the one corresponding to the prosody embedding.

For the Styles dataset, the technique mostly fails, resulting in the same prosody with any of the AR prior model's top predictions. Only for a few phonemes, a difference was found but limited to the length of the phoneme. This can be explained by the lack of variety within styles in this dataset.

5. CONCLUSION

This paper proposed a novel model for multi-level controllable speech synthesis, using a reference encoder for the global style and a VQ-VAE for fine-grained prosody. Furthermore, we proposed an interactive method for editing the local prosody, by choosing new embeddings from the top predictions of an autoregressive prior, trained over the fine-grained prosody embeddings. Experimental results showed that a global prosody embedding from a reference speech sample could be used to control the speaking style of the output speech, and that the changing the fine-grained prosody embeddings indeed lead to a change in local prosody.

[2]The LibriTTS corpus is multi-speaker and can thus be used as a general vocoder. However, since LibriTTS does not include the voices of "Styles" and "Blizzard", the generated audio samples contain some voice distortion.

6. REFERENCES

[1] G. Sun, Y. Zhang, R. J. Weiss, Y. Cao, H. Zen, A. Rosenberg, B. Ramabhadran, and Y. Wu, "Generating diverse and natural text-to-speech samples using a quantized fine-grained VAE and autoregressive prosody prior," in *ICASSP*, 2020.

[2] I. Elias, H. Zen, J. Shen, Y. Zhang, Y. Jia, R. J. Weiss, and Y. Wu, "Parallel tacotron: Non-autoregressive and controllable TTS," in *ICASSP*, 2021.

[3] CM Chien and HY Lee, "Hierarchical prosody modeling for non-autoregressive speech synthesis," *CoRR*, vol. abs/2011.06465, 2021.

[4] RJ Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron," in *ICML*, 2018, vol. 80.

[5] Y Wang, D. Stanton, Y. Zhang, R. J. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *ICML*, 2018.

[6] YJ Zhang, S. Pan, L. He, and Z. Ling, "Learning latent representations for style control and transfer in end-to-end speech synthesis," in *ICASSP*, 2019.

[7] Y. Lee and T. Kim, "Robust and fine-grained prosody control of end-to-end speech synthesis," *CoRR*, vol. abs/1811.02122, 2018.

[8] WN Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen, P. Nguyen, and R. Pang, "Hierarchical generative modeling for controllable speech synthesis," *CoRR*, vol. abs/1810.07217, 2018.

[9] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and TY Liu, "FastSpeech 2: Fast and high-quality end-to-end text to speech," 2020.

[10] G. Zhang, Y. Qin, and T. Lee, "Learning Syllable-Level Discrete Prosodic Representation for Expressive Speech Generation," in *Interspeech*, 2020.

[11] A. Lańcucki, "FastPitch: Parallel text-to-speech with pitch prediction," *CoRR*, vol. abs/2006.06873, 2021.

[12] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and TY Liu, "FastSpeech: Fast, robust and controllable text to speech," in *NeurIPS*, 2019.

[13] Y. Wang, R.J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Interspeech*, 2017.

[14] D. Griffin and Jae Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1984.

[15] A. van den 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," *CoRR*, vol. abs/1609.03499, 2016.

[16] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *ICML*, 2018.

[17] JM Valin and J. Skoglund, "Lpcnet: Improving neural speech synthesis through linear prediction," in *ICASSP*, 2019.

[18] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP*, 2020.

[19] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NeurIPS*, 2014, NeurIPS.

[20] C. Miao, S. Liang, M. Chen, J. Ma, S. Wang, and J. Xiao, "Flow-tts: A non-autoregressive network for text to speech based on flow," in *ICASSP*, 2020.

[21] K. Peng, W. Ping, Z. Song, and K. Zhao, "Non-autoregressive neural text-to-speech," in *ICML*, 2020.

[22] GE Henter, J. Lorenzo-Trueba, X. Wang, and J. Yamagishi, "Deep encoder-decoder models for unsupervised learning of controllable speech synthesis," *CoRR*, vol. abs/1807.11470, 2018.

[23] G. Sun, H. Zen, R. J. Weiss, Y. Wu, Y. Zhang, and Y. Cao, "Fully-hierarchical fine-grained prosody modeling for interpretable speech synthesis," in *ICASSP*, 2020.

[24] T. Raitio, R. Rasipuram, and D. Castellani, "Controllable neural text-to-speech synthesis using intuitive prosodic features," *Interspeech*, 2020.

[25] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *NeurIPS*, 2017.

[26] Simon King and Vasilis Karaiskos, "The Blizzard Challenge 2013," in *Proceedings Blizzard Workshop 2013*, Sept. 2013.

[27] T. Hayashi, R. Yamamoto, K. Inoue, T. Yoshimura, S. Watanabe, T. Toda, K. Takeda, Y. Zhang, and X. Tan, "Espnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit," in *ICASSP*, 2020.

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

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.

[30] H. Zen, R. Clark, R. J. Weiss, V. Dang, Y. Jia, Y. Wu, Y. Zhang, and Z. Chen, "Libritts: A corpus derived from LibriSpeech for text-to-speech," in *Interspeech*, 2019.

[31] N. Tits, *Controlling the Emotional Expressiveness of Synthetic Speech - a Deep Learning Approach*, Ph.D. thesis, Université de Mons, 12 2020.