

# Supplementary Material

## No Gestures Left Behind: Learning Relationships between Spoken Language and Freeform Gestures

Chaitanya Ahuja, Dong Won Lee, Ryo Ishii, Louis-Philippe Morency

Language Technologies Institute

Carnegie Mellon University

Pittsburgh, PA

{cahuja, dongwonl, rishii, lmorency}@cs.cmu.edu

### 1 More analyses

**Weight updates on a toy example** For a toy example that learns to generate samples from a gaussian distribution, we show how the resampling weights look like for each bin over the course of optimization in Figure 1. We also note that AISLe recognizes discrepancies throughout the proposal distribution and assigns larger weights (local maxims denoted by  $\text{—}$ ) to the samples in those bins. Initially, the weights are almost equal for all bins, because the discriminator is still learning to predict the correct likelihood. As the training progresses, the network learns to first focus on the center of the distribution, before shifting gears towards the tail close to the end of training.

#### Coverage vs frequency of occurring words

While it is expected that rarely occurring words will be less likely to generate gestures that represent the true distribution (right half of Figure 2), we see that AISLe is able to push boundaries of MMS-Transformer to generate more correct distributions for words in the long tail.

**More qualitative analysis** We compare the coverage of the generated gestures for our model and baselines in Figure 9. We also show generated gesture as a skeleton plot over the ground truth and compared with previous work in 8. While, these images give some idea about the qualitative performance, we would recommend looking at the attached video for a better understanding.

**Speaker-wise objective results** We also have the speaker-wise objective results in Figures 10-14.

### 2 Model

#### 2.1 Estimating Mixture Model Priors during Training

During training, we partition poses  $Y_p$  into  $M$  clusters using an unsupervised approach, Lloyd’s algorithm (Lloyd, 1982). While other unsupervised clustering methods (Reynolds, 2009) can also be used at this stage, we choose Lloyd’s algorithm for its simplicity and speed. Each of these clusters represent samples from probability distributions  $\{p^1(y|x), p^2(y|x), \dots, p^M(y|x)\}$ . If a sample belongs to the  $m^{\text{th}}$  cluster,  $\phi_m = 1$ , otherwise  $\phi_m = 0$ , making  $\Phi$  a sequence of one-hot vectors. While training the generator  $G_\theta$ , if a sample belongs to the distribution  $p^m(y|x)$ , only parameters of sub-generator  $G_m$  are updated. Hence, each sub-generator learns different components of the true distribution, which are combined using Equation 9 (main paper) to give the generated pose.

#### 2.2 Aligning Multi-Scale Embeddings

As language and audio have different scales, we augment the idea of positional embeddings proposed in (Vaswani et al., 2017) to provide the information of word-level ordering as well as sub-word frame-level ordering.

**Word-level Ordering:** Given language embeddings  $\mathbf{Z}^w \in \mathcal{R}^{N \times h^w}$ , where  $N$  represents the number of words in a sampled sequence, and  $pos \in N$  is the dimensional position of the word in the sequence. The term  $i \in h^w$  represents the  $i$ -th position word embedding and is used to ensure that each positional encoding corresponds to a sinusoid. We add the corresponding word-level positional embedding for each word embedding.

The positional embedding is derived as the following:

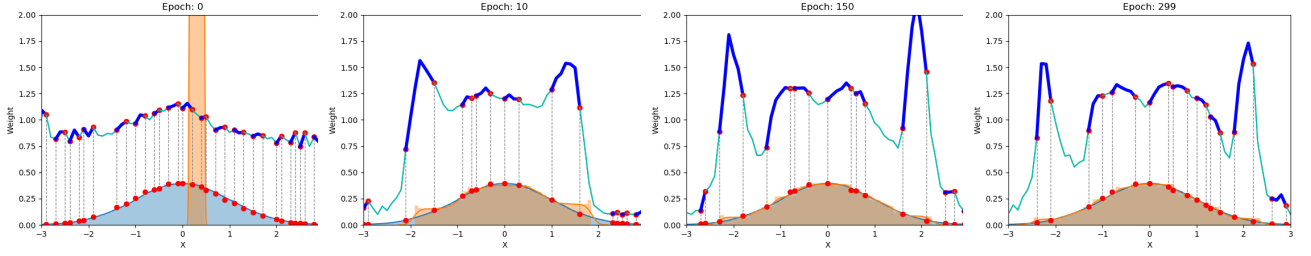


Figure 1: Progress of learning a Gaussian distribution using AISLe on top of a vanilla GAN. The top line plot refers to the weights assigned adaptively assigned to the samples corresponding to the bins on the X axis. Initially, the model focuses on the samples close around zero and gradually moves on to focusing on the heavy tail of the distribution. The dark-blue segments of the line plot refer to local maximas and segments that require more attention. The dotted vertical lines correspond to inflection points of the weight line plot.

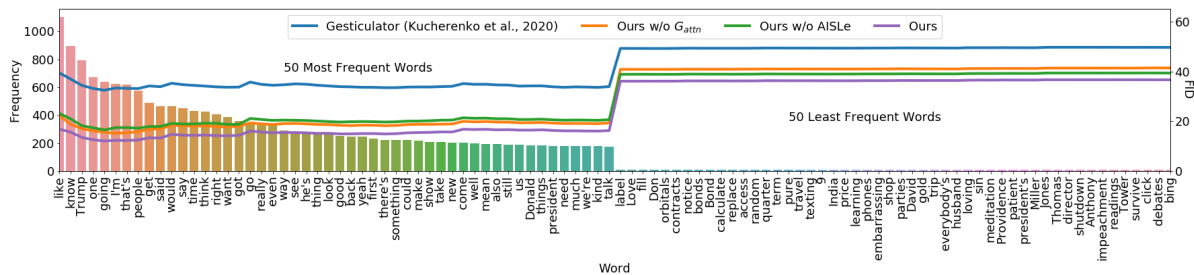


Figure 2: Words vs FID. The top 100 occurring words are shown

$$\mathbf{PE}_{pos,i} = \sin(pos/10000 \frac{2i}{h^w}), \quad \text{if } i \text{ even} \quad (1)$$

$$\mathbf{PE}_{pos,i} = \cos(pos/10000 \frac{2i-1}{h^w}), \quad \text{if } i \text{ odd} \quad (2)$$

**Frame-level Ordering:** Given a single language embeddings  $\mathbf{Z}^w \in \mathcal{R}^{N \times h^w}$ ,  $\mathbf{Z}^w$  occupies multiple time-frames. In order to account for the frame-wise progression of the word, we use the same positional embedding as shown in the above equation. We additionally process the word duration of each word, which represents the number of frames each word occupies. Then, we replace  $pos \in \text{Word Duration}$  with the position of the frame for the word. We add the corresponding frame-wise positional embedding for each frame-level word embedding.

### 3 Experiments

#### 3.1 Baselines

**Gesticulator(Kucherenko et al., 2020):** Unlike MMS-Transformer, Gesticulator is an autoregressive model for generating gestures using text and speech. The audio inputs are represented via log-mel-spectrograms. For text features, in comparison

to multi-scale BERT embeddings used for MMS-Transformer, single scale BERT embeddings are used for the Gesticulator. The text features are repeated to align with audio frames. Furthermore, text features corresponding to filler words and features for silence, which do not contain semantic information, is additionally processed. Using WebRTC Voice Activity Detection (WebRTC) to find timesteps with silence, all elements of the in the embeddings corresponding to silence is set to -15 and made distinct from all other audio encodings. Additionally, filler words are found for each speaker’s transcripts using the NLTK package. Then, the weighted average of the BERT embeddings of all filler words spoken per speaker are calculated. The averaged filler BERT embedding replaces each filler word spoken by the speaker. After processing the data to account for silence and filler words, in order to provide more contextual information, a sliding window of audio features, including 7 and 15 future time steps are concatenated with the current time step features (audio and text) to produce a long vector.

In comparison to MMS-Transformer, implemented with cross-modal multihead attention, CNNs and adversarial training, the Gesticulator’s

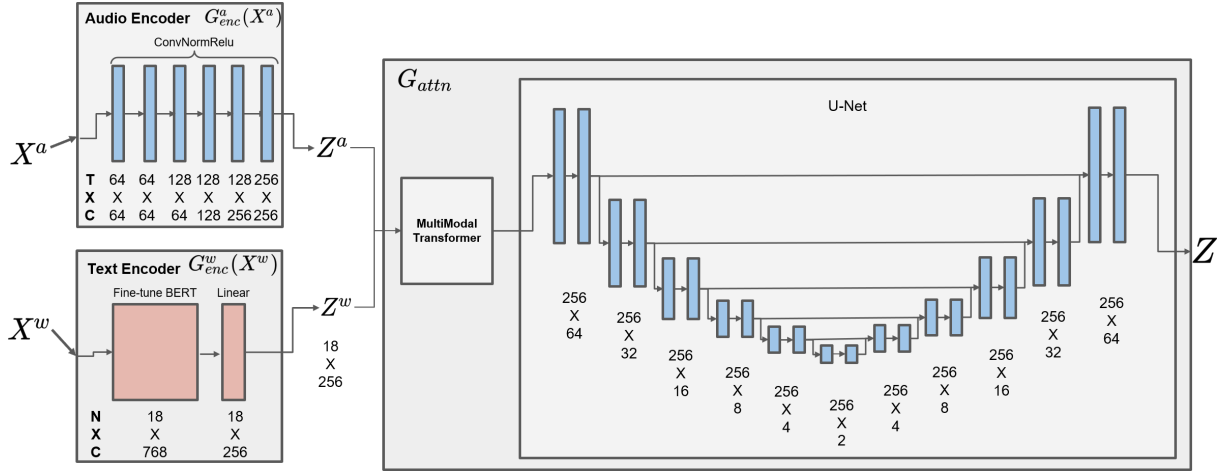


Figure 3: Encoder Architecture

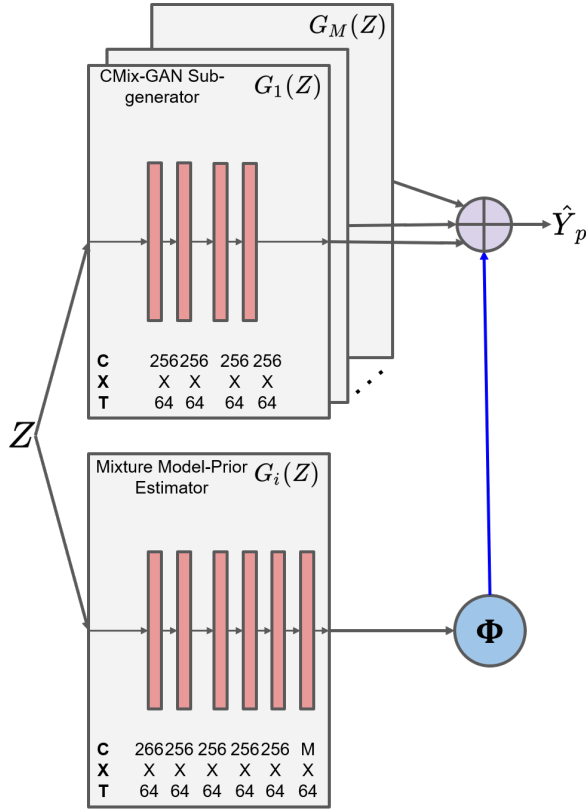


Figure 4: Decoder Architecture

model architecture relies solely on fully connected layers. Given the processed input as described above, 3 fully connected layers are applied to reduce dimensionality and producing an output  $x$ . Furthermore, unlike our model, autoregression is applied via FiLM conditioning, where previous 3 poses are taken as input and fed into fully connected layers to produce scaling  $\alpha$  and offset vectors  $\beta$ . Then the output is applied to element-wise affine transformations:  $x * \alpha + \beta$ . For the first 7

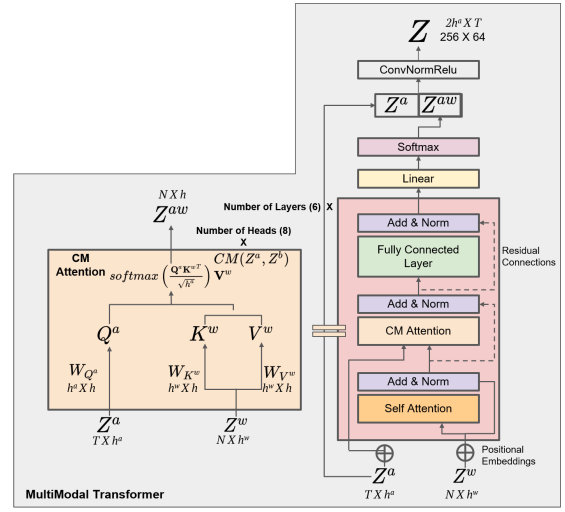


Figure 5: MultiModal Transformer Architecture

epochs, no FiLM conditioning is applied. Then for the proceeding 5 epochs, varying teacher forcing is applied where the number of times the model receives ground-truth poses is annealed over time. By the 12th epoch, the model uses its own generated poses in FiLM conditioning (Perez et al., 2018). Finally, the loss function is a sum of MSE between the poses and the velocities of the gestures.

### 3.2 Implementation details

We use PyTorch as the auto-differentiation library to train all our models. The detailed description of our model, with layer sizes, is described in Figures 3, 4 and 5.

In our experiments, we use the following hyperparameter settings: Our batch size is 32, sampling intervals of approximately 4.0 seconds for each batch. We use a overlapping windows during

sampling with step-size of 5.

In order to find the optimal learning rate within the range of 0.00001 to 0.00005, we uniformly sampled with an increment of 0.00001 and ran an hyperparameter search on one model for one speaker. We found that the learning rate 0.00003 was marginally better than the others, making it our choice for all models. Furthermore, in training, we use Adam with rectified weight decay (Loshchilov and Hutter, 2017) with a linearly decaying learning rate schedule.

The number of training iterations are 40000 and we check the validation score at every 400 iterations, making sure that the model runs for a minimum of 20000 iterations before it considers early stopping.

We use  $M=8$  for the mixture of GANs. We choose 8 by running an ablation which shows that the performance plateaus after 8.

The average model train runtime was around 24 hours (+ 6 hours if it decided to run the complete 40000 iterations) on Titan X 1080 GPUs.

The following evaluation metrics were used, with links provided:

- FID: [https://github.com/mseitzer/pytorch-fid/blob/master/fid\\_score.py](https://github.com/mseitzer/pytorch-fid/blob/master/fid_score.py)
- WD1: [https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wasserstein\\_distance.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wasserstein_distance.html)
- PCK: <https://github.com/amirbar/speech2gesture/blob/master/common/evaluation.py>

## 4 PATS Dataset

Most of the details and pre-processing on the dataset can be found in Section 5.3.1 of the main paper. we used a train/validation/test split of 80/10/10 which is fixed to ensure consistency across experiments. A visual description of the dataset, which compares the lexical and gesture diversity of each individual speaker, can be found in Figure 6. Link to dataset: <http://chahuja.com/pats>

### 4.1 Human Perceptual Study

We attach a screenshot of a sample study and the questions asked to the users in Figure 7

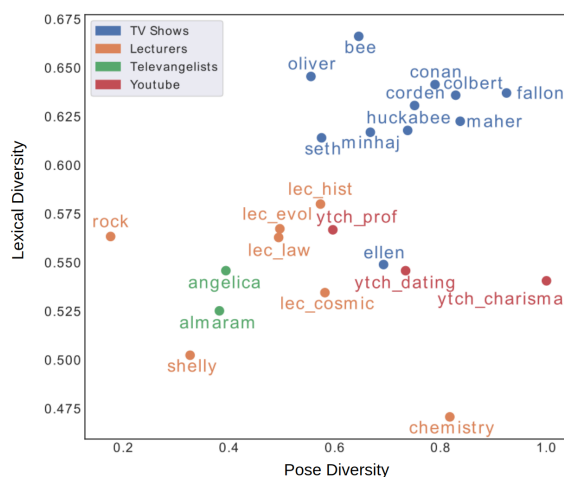


Figure 6: A visual representation of all speakers in PAT+ dataset. X-axis represents the average diversity of gestures while Y-axis denotes the lexical diversity in the speakers transcripts.

## References

- Taras Kucherenko, Patrik Jonell, Sanne van Waveren, Gustav Eje Henter, Simon Alexanderson, Iolanda Leite, and Hedvig Kjellström. 2020. Gestulator: A framework for semantically-aware speech-driven gesture generation. *arXiv preprint arXiv:2001.09326*.
- Stuart Lloyd. 1982. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2018. Film: Visual reasoning with a general conditioning layer. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Douglas A Reynolds. 2009. Gaussian mixture models. *Encyclopedia of biometrics*, 741.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- WebRTC. “webrtc,” 2017. [online]. available: <https://webrtc.org/>.

amazonmturk  
Worker

**Instructions** ×

[View full instructions](#)

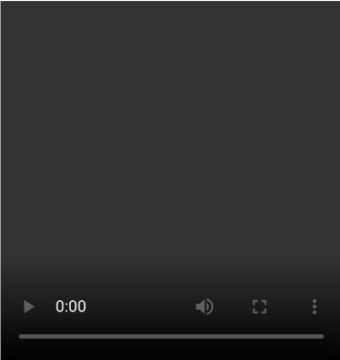
Please read all instructions carefully before starting. There are a lot of HITs in this task with the same instructions. Hence as a one-time investment, I would urge you to understand the task before proceeding with the annotation. Please feel free to contact me if you have any questions.

Both videos have the same audio segment. The video is an animation of the speaker corresponding to the audio segment.

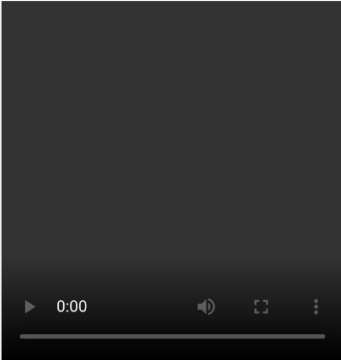
- Turn on the speakers or use headphones as the videos have audio.
- See both videos one by one.
- Choose the appropriate answers to the questions about the animations you have just seen

**Definitions**

- Timing:** People tend to emphasize with their hand gestures when they emphasize what they are saying. Timing is best when the gestures align (i.e., occur simultaneously) with the relevant spoken words. These two events occur simultaneously for the timing to be correct.
- Expressiveness:** Expressiveness is a general measure of the amount of gestures. It is not only about the number of gestures but also about the size of these gestures. More and larger gestures will represent more expressiveness.
- Relevant:** The form of the gesture should not only be well timed (as judge with the Timing metric) but also seem to be the right gesture, relevant with the spoken words. For example, if a person says "me", and simultaneously point towards themselves, then the gesture is relevant.
- Naturalness:** This is a general metric which asks you to judge if the animation looks natural, as if it was the depiction of a real person. The naturalness involves both the body and gestures, as well as how they appear in relation with the spoken words. The gestures need to look natural.



**Video A**



**Video B**

Which animation has the best **Timing** of gestures with respect to the spoken words?

A  Neutral  B

Which animation has most **Relevant** gestures with respect to the spoken words?

A  Neutral  B

Which animation has the most **Expressive** gestures?

A  Neutral  B

Which animation looks the most **Natural**, with natural-looking gestures?

A  Neutral  B

Figure 7: Screenshot of MTurk Experiment used to measure subjective metrics

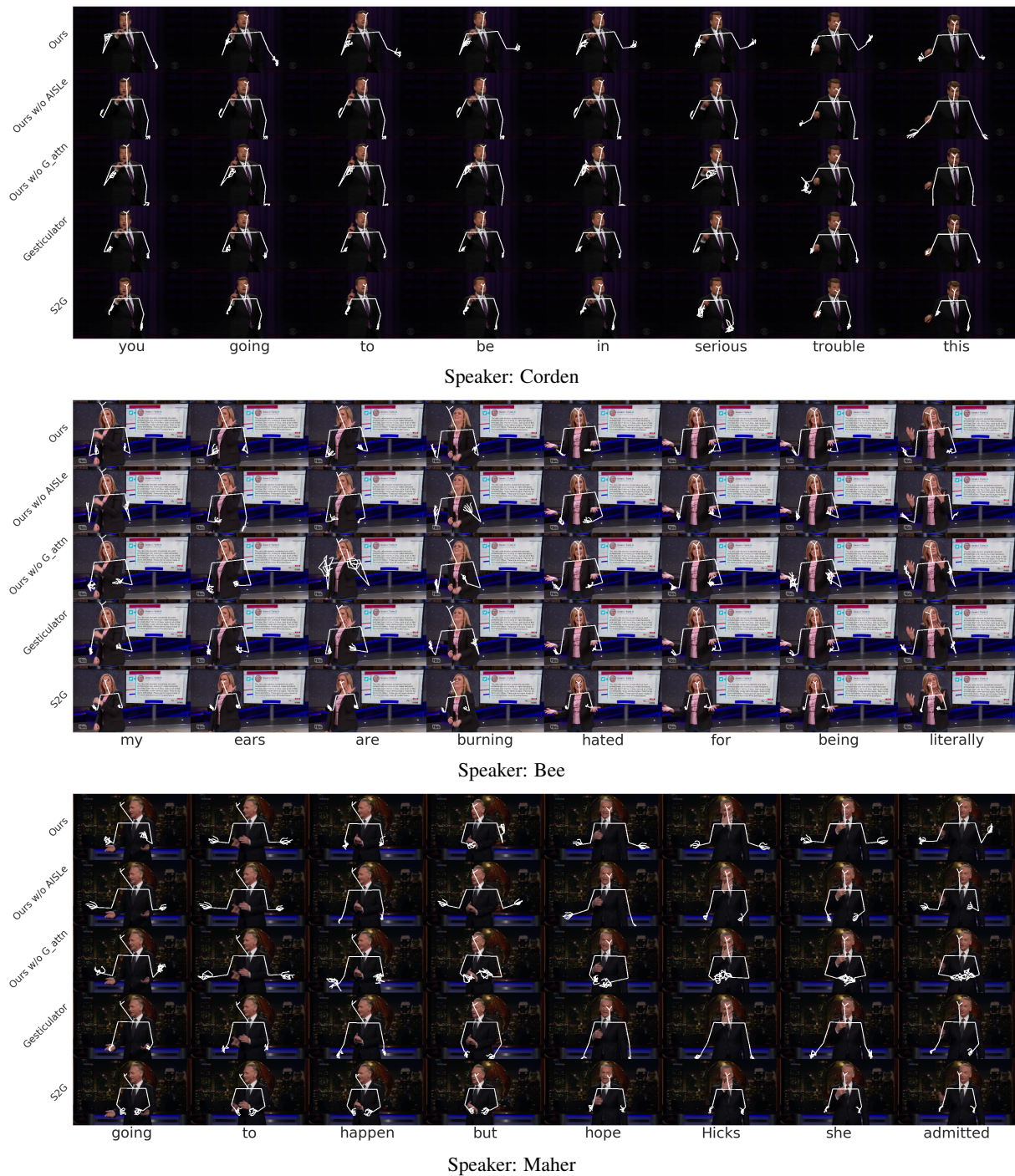


Figure 8: Generated animations are plotted as frames over the ground truth video frames. The text at the bottom refers to the context of the generation. While, these images give some idea about the qualitative performance, we would recommend looking at the attached video for a better understanding.

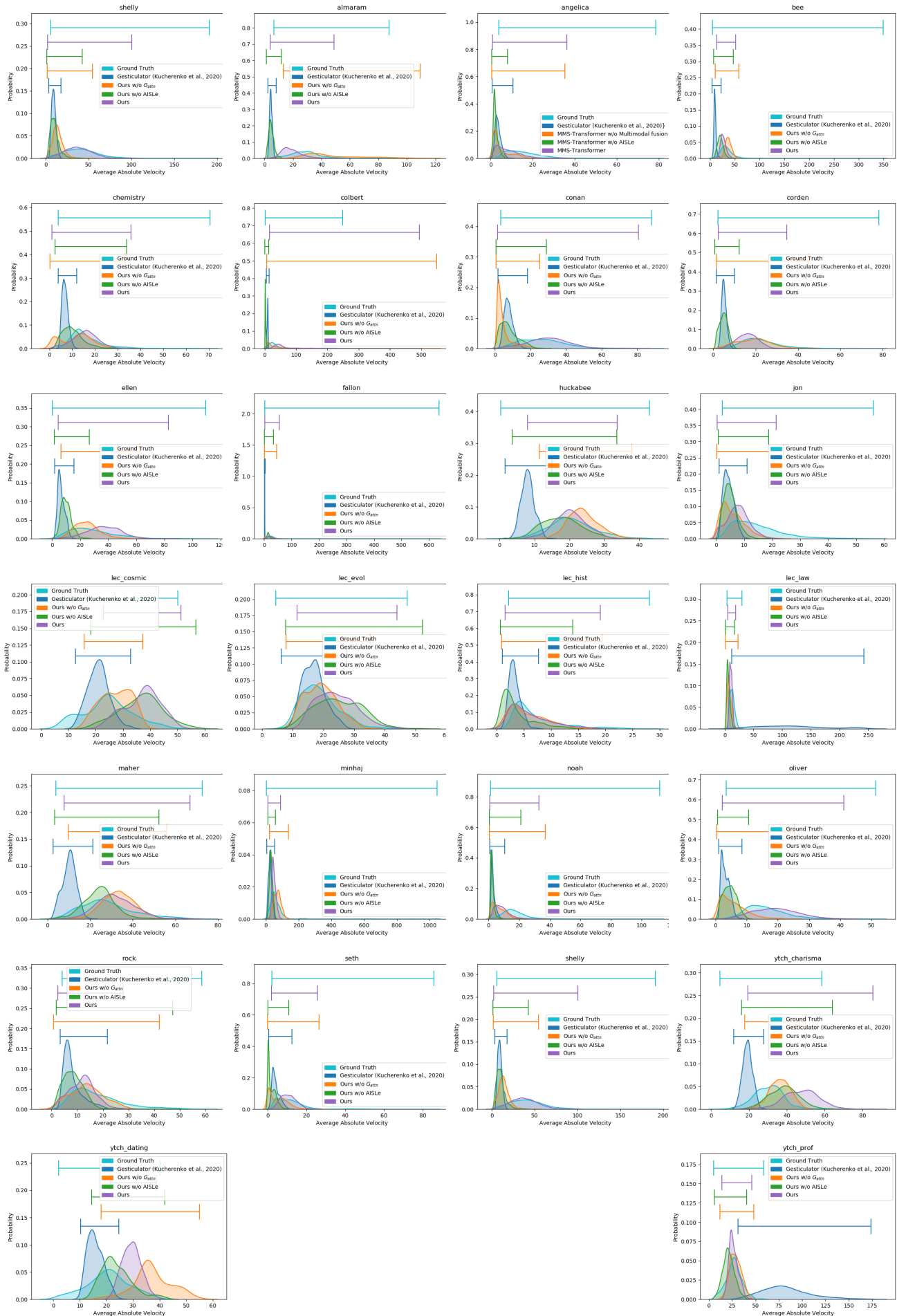


Figure 9: Distribution of the generated gestures with average absolute velocity as the statistic for four different speakers. The support (or coverage) of the distribution is denoted with the colour coded lines at the top of each plot. Larger overlap of a model’s distribution with the ground truth distribution is desirable.

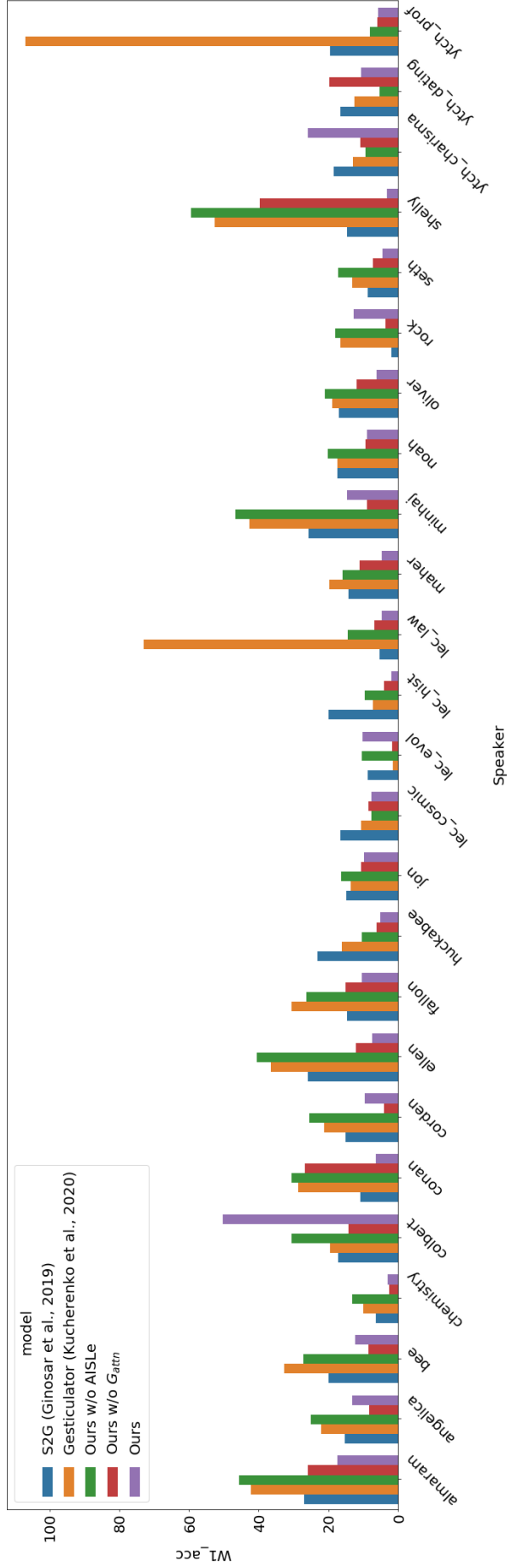
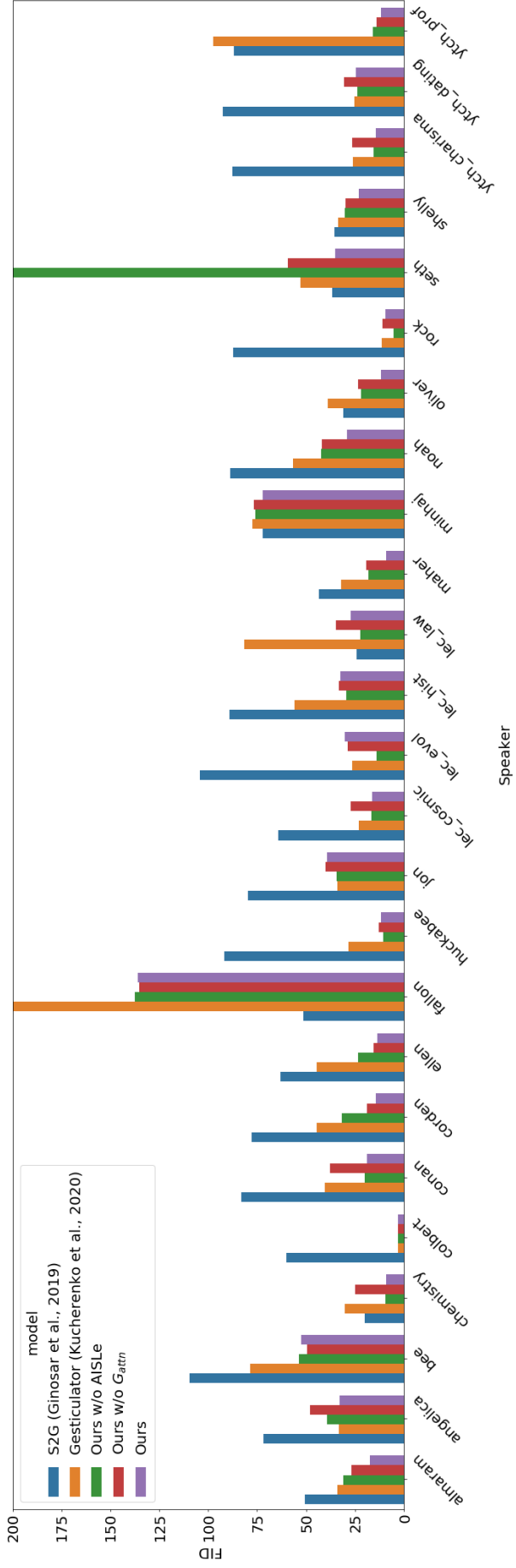


Figure 11: W1 (acc.)



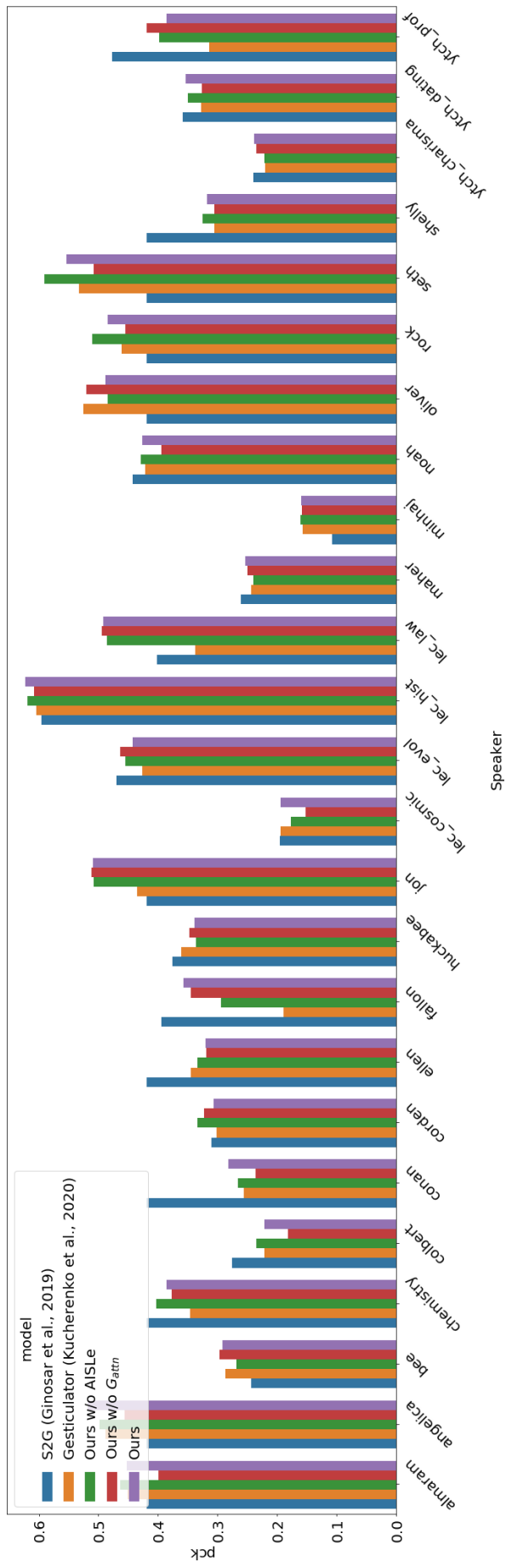
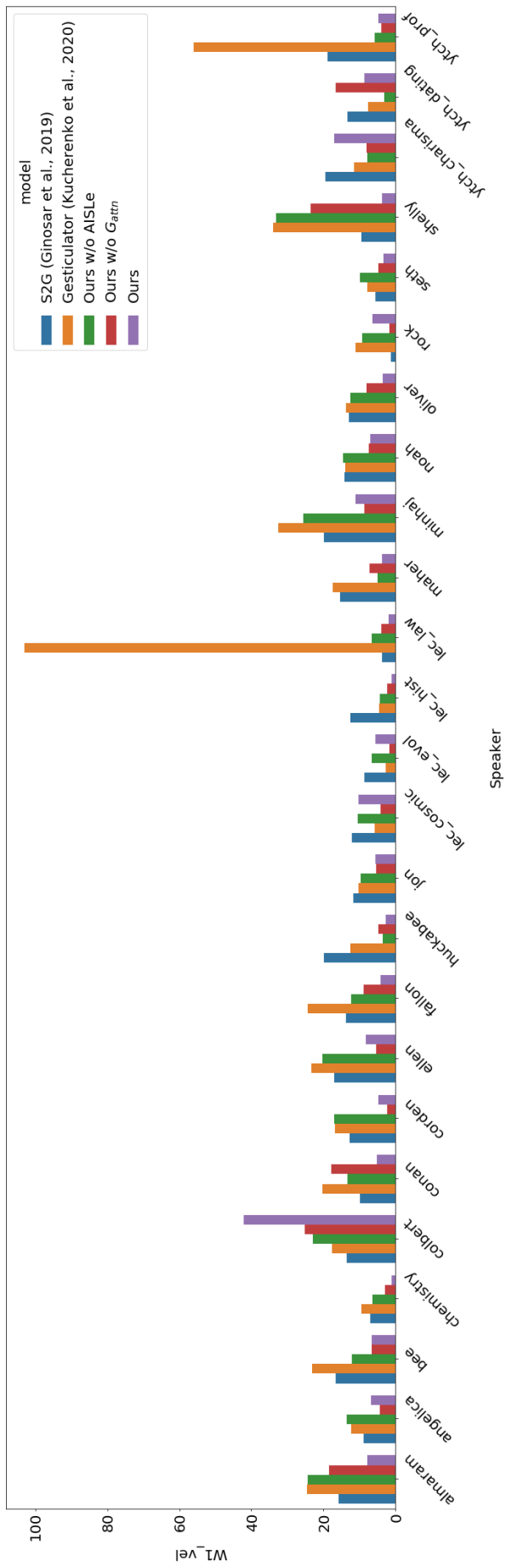


Figure 13: PCK

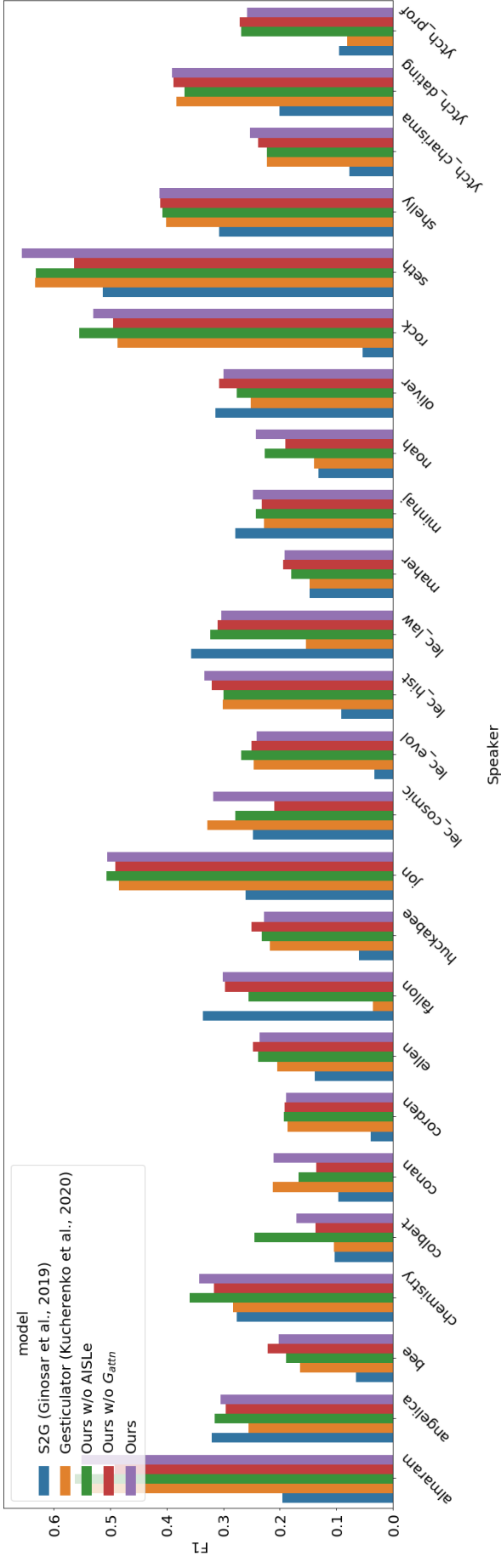


Figure 14: F1