INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Effects of Feature Scaling and Fusion on Sign Language Translation

Tejaswini Ananthanarayana, Lipisha Chaudhary, Ifeoma Nwogu

Rochester Institute of Technology, USA

ta2184@rit.edu, lc2919@rit.edu, ionvcs@rit.edu

Abstract

Sign language translation without transcription has only recently started to gain attention. In our work, we focus on improving the state-of-the-art translation by introducing a multi-feature fusion architecture with enhanced input features. As sign language is challenging to segment, we obtain the input features by extracting overlapping scaled segments across the video and obtaining their 3D CNN representations. We exploit the attention mechanism in the fusion architecture by initially learning dependencies between different frames of the same video and later fusing them to learn the relations between different features from the same video. In addition to 3D CNN features, we also analyze pose-based features.

Our robust methodology outperforms the state-of-the-art sign language translation model by achieving higher BLEU 3 - BLEU 4 scores and also outperforms the state-of-the-art sequence attention models by achieving a 43.54% increase in BLEU 4 score. We conclude that the combined effects of feature scaling and feature fusion make our model more robust in predicting longer n-grams which are crucial in continuous sign language translation.

Index Terms: sign language translation, transformer, attention, multi-feature.

1. Introduction

Research on automated sign language understanding has been limited, especially when compared with its spoken language counterpart. Sign language interpretation can be challenging due to the lack of easy accessibility to human interpreters - sign language interpreters can be expensive and not readily available. Automating the process of sign language translation (SLT) can therefore greatly facilitate the communication between signing and non-signing people in the community.

Sign language typically comprises of hand movements, facial movements/ expressions, body movements, location references, and sometimes lip movements [1, 2]. The unit components of sign language are not directly translatable to their spoken word counterparts; rather, signs are directly related to an intermediary symbolic language called *gloss*. Gloss can be viewed as the written form of sign language and is useful for transcription. More information on gloss is provided in Section 2. As glosses align directly with signs (unlike spoken words), initial studies in automated SL understanding required them as an intermediate step in performing translation, a problem we refer to more appropriately as recognition or transcription. The availability of gloss simplifies the task of SL understanding, as the model first performs gloss-based recognition and as a secondary step, translates the gloss to text [3]. Alignment-based loss functions such as the connectionist temporal classification (CTC) loss are used and they go a long way in facilitating the process.

But for real-time sign language understanding in real-world situations, where gloss is unavailable, it is imperative to design SL systems that can bypass the need for gloss during translation. We propose such a system where we directly perform continuous sign language translation (SLT) without gloss, a significantly more challenging task.

In this work, we aim to study the effects of feature scaling and feature fusion for SLT by introducing a multi-feature fusion attention architecture. As sign language is hard to segment, we perform feature scaling as a pre-processing step to find the boundaries of the signs.

Our specific contributions are listed below:

- We present an SLT framework that uses a feature scaling mechanism to implicitly locate sign boundaries, thus improving the overall SLT mechanism. We also extract scaled, pose-based features (explained in more detail in Section 5). Similar to word boundary segmentation in speech, sign segmentation is a challenging problem and the use of feature scaling facilitates this.
- We develop a novel multi-feature fusion architecture, which implements self-attention on individual scaled features by attending to information between different blocks of frames. The self-attention from each of the individual features is then fused and passed to an intermediary multi-feature attention block that relates all the features (explained in more detail in Section 6).
- We perform ablations, which show that our proposed multi-feature fusion architecture outperforms the current state-of-the-art results, when applied on the well-known RWTH-PHOENIX-Weather SL dataset [3] (explained in more detail in Section 7).

2. Sign Language Transcription - Gloss

The transcribed form of sign language is commonly referred to as *glossing* [4, 5]. The process of first transcribing visual sign language to its exact symbolic gloss form and then extending this to the spoken language form has been performed by deep learning researchers [3]. We describe this process as SL recognition or transcription.

Glosses can differentiate between finger-spelling and complete word signs. For example, most proper nouns like names of people are often finger-spelled and this can be written symbolically as *fs*. Other constructs of gloss include *LOC* indicating location, ++ indicating a repetition of sign, and repeated references to a pre-defined multi-dimensional space with some specific meaning. For example, a signer can assign a 3D space to a particular construct, "Jason", and refer to that space whenever the signer wants to mention "Jason", as a way to avoid to finger-spelling every time. Gloss readily captures such complex constructs.

Annotating SL via glossing requires specialized skills and can be expensive. SL understanding can become a very challenging task if it solely relies on gloss. Much of the current computational research work in SL understanding [3, 6] use gloss as

http://dx.doi.org/10.21437/Interspeech.2021-1863

an intermediate step before converting to spoken language text. However, work done using gloss cannot be easily extended to datasets where gloss is not available, specifically when we target low-resource sign languages.

For this reason, the main focus of this paper is to perform continuous SL translation without using any prior transcriptions. Our work outperforms current state-of-the-art results for SL translation without gloss.

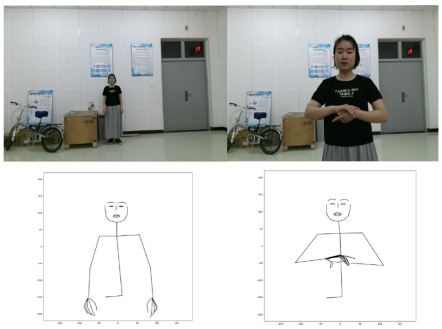

Figure 1: *Canonical OpenPose representation.*

3. Related Works

Sign language translation can be loosely categorized into word level and sentence level translations. Most of the word level translations involve classification of signs or in other words action/ gesture recognition [7, 8, 9, 10, 11, 12, 13]. Word-level classification or recognition is a simpler task when compared to continuous sign translation.

Gloss based SL recognition falls into the classification category. This is usually performed using the CTC loss [3, 14, 15]. Glosses are also used to pretrain a network like a CNN-LSTM-HMM (Convolution Neural Network – Long Short-Term Memory – Hidden Markov Model) model which aligns the glosses with the input frames. After training, new frames can be passed through the pre-trained CNN to obtain new set of features that can be fed in as input to the translation model [6]. Very recently researchers are bringing their attention to this problem [16] and focusing on improving SLT models when gloss is not available.

SLT architectures have evolved over time and borrowed inspiration from many state-of-the-art sequence-to-sequence networks [17, 18, 19], sequence-to-sequence with attention [20, 21, 3]. Most of the recent architectures leverage the attention mechanism in the transformer model due to its capability to predict longer tokens than typical sequence networks [22, 23, 24, 25, 26, 14, 15].

4. Dataset

The publicly available German RWTH-PHOENIX-Weather dataset [3] is collected from weather forecast airings. We use 7096/519/642 train/val/test samples. The original resolution of frames is 210×260 with each video recorded at 25 frames per second. The dataset was signed by 9 individuals.

5. Input features

5.1. Pose features

We extract pose features from OpenPose [27], an open-source toolkit for extracting body keypoints. We obtain x, y locations for 25 body-joint keypoints, 21 keypoints for each hand, and 70 facial landmark keypoints, resulting in 274 points per frame. We keep the order of the joints the same as the original authors [27]. For frames where the lower body information is not available, the landmarks are zeroed out. We obtain a canonical form of the frame keypoints, by centering all points to the origin and scaling them to the same size. Examples of the canonical form of OpenPose points is shown in Figure 1 [1].

5.2. Scaled Features

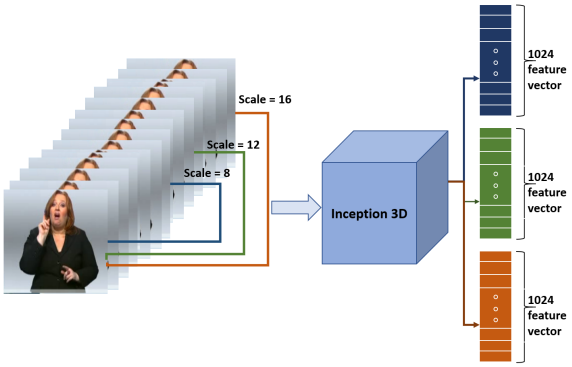

Figure 3: *3D CNN Feature scaling. Scale 8 (blue), Scale 12 (green), Scale 16 (orange) are shown (best viewed in color).*

As sign language is challenging to segment, we aim to find implicit boundaries of signs by scaling the input frames. We apply a sliding window encompassing multiple frames across the video, resulting in overlapping video segments. The length of the sliding window can be 8, 12, or 16. The frames in each video segment is passed to the pre-trained Inception 3D CNN [28] model to obtain an output 1024 feature vector. This vector (or CNN embedding) is obtained for each video segment as shown in Figure 3. The features obtained from each of these three scaling mechanisms are fed to the multi-feature fusion model as shown in Figure 2.

Consider X_1, X_2, X_3 to be the features processed by CNN_8, CNN_{12} and CNN_{16} as shown in Equation (1); where CNN_m is the i3D pre-trained CNN network whose inputs take m-frames.

$$\begin{aligned} X_1 &\in \text{CNN}_8\{x_1, x_2, .., x_n\} \\ X_2 &\in \text{CNN}_{12}\{x_1, x_2, .., x_n\} \\ X_3 &\in \text{CNN}_{16}\{x_1, x_2, .., x_n\} \\ X_4 &\in \text{OP}_8\{x_1, x_2, .., x_n\} \end{aligned} \tag{1}$$

Each x_i in $x_1, x_2, .., x_n$ represents a group of frames; i.e. x_1 for scale 8 is equivalent to frames $[0, 1, ...7]$, x_2 is equivalent to frames $[2, 3, ...9]$ and so on. For OpenPose we take the center frame which is a good representation for the whole segment.

[1] These sample frames are not from the GSL dataset described in this paper.

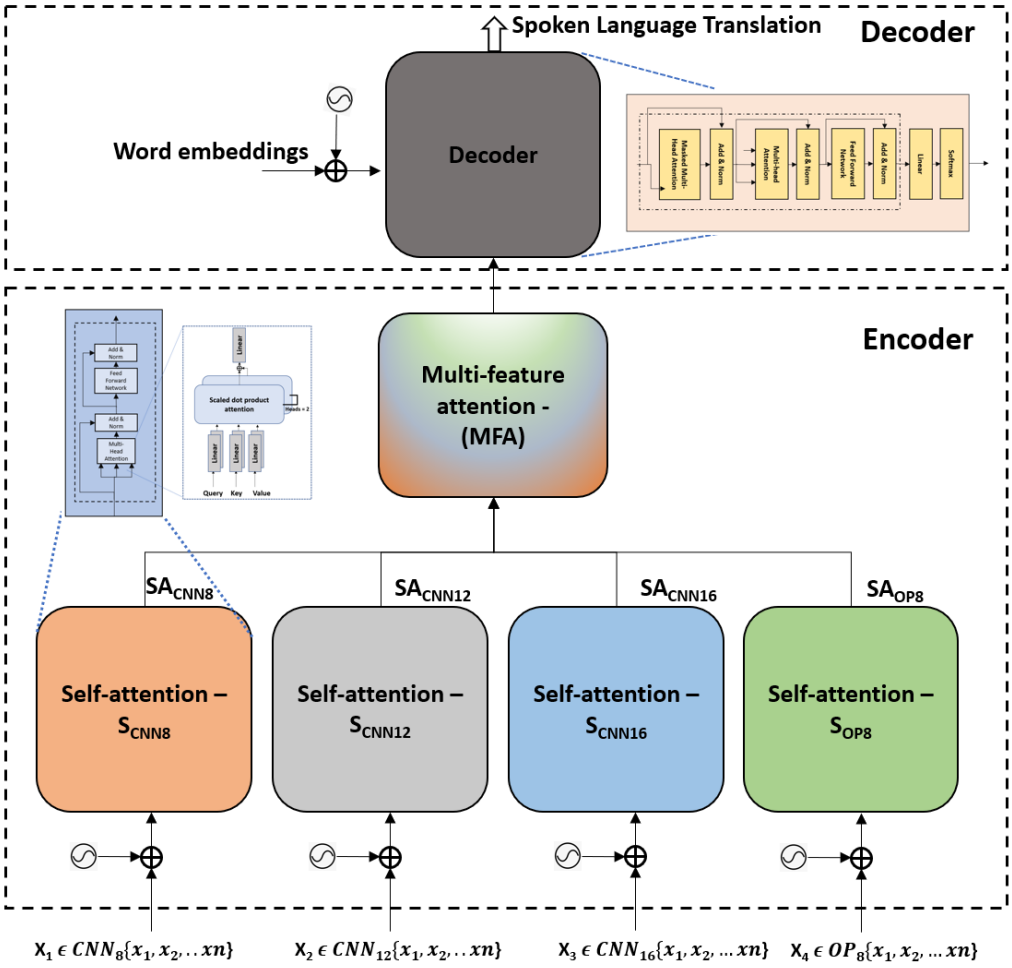

Figure 2: *Multi-feature fusion architecture for sign language translation (best viewed in color). (Notations are described in Section 6.)*

Having overlapping frames in the segment is important, as the model learns better from the repetition of signs. Mirror padding is applied wherever needed, so that the total number of frames for each video is the same for all three scales.

6. Methodology

To perform SLT, we introduce a multi-feature fusion architecture, shown in Figure 2. The model performs self-attention on individual features. The attention from individual features is then fed as input to the multi-feature attention (MFA) module which performs fused attention. The fused attention obtained from the encoder portion is then passed on to the decoder. The decoder takes as input the word embeddings, specifically the words predicted from previous time steps. The encoder output is used in the encoder-decoder attention block to attend to information between the visual features + pose and word embeddings. The decoder then outputs spoken language translation.

The subsections below describe various blocks of the multi-feature fusion architecture in detail.

6.1. Encoder

The pose and visual features obtained from the scaling mechanism are shown in Equation (1). Here, X_1, X_2, and X_3 are features obtained through the implicit boundary segmentation processed by CNN_8, CNN_{12}, CNN_{16}, respectively.

Positional encoding is added to the input embeddings before passing it to the self-attention blocks. The positional encoding is essential to maintain the order information.

Each of the self-attention blocks ($S_{CNN8}, S_{CNN12}, S_{CNN16}, S_{OP8}$) compute the importance of each frame with respect to all the frames by initially performing a dot product between them (Q_{OP8}, K_{OP8}). The Softmax obtained from the dot product is then used to calculate the actual weight that each frame has in the original input by multiplying it again with the input frames (V_{OP8}). This attention calculation for one of the blocks S_{OP8} is mathematically described in Equation (2).

$$\text{Self Attention } (SA_{OP8}) = \text{Softmax}(\frac{Q_{OP8}K_{OP8}^T}{\sqrt{d_k}})V_{OP8} \tag{2}$$

This attention is then passed on to a feed-forward network which learns two linear layers. Similar to the original implementation [22] we retain the add and norm blocks, and, skip connections.

The output from individual self-attention blocks are then fused

Table 1: *Results on the GSL dataset using the multi-feature fusion architecture*

Architecture	**BLEU 1**	**BLEU 2**	**BLEU 3**	**BLEU 4**
Conv2d-RNN [3]	27.10	15.61	10.82	8.35
Conv2d-RNN [3] + Luong Attention [21]	29.86	17.52	11.96	9.00
Conv2d-RNN [3] + Bahdanau Attention [29]	32.24	19.03	12.83	9.58
TSPNet-Sequential [16]	35.65	22.80	16.60	12.97
TSPNet-Joint [16]	**36.10**	**23.12**	16.88	13.41
Transformer (CNN_8) (Ours)	25.22	16.29	11.87	9.31
Transformer (CNN_{12}) (Ours)	24.48	16.01	11.71	9.31
Transformer (CNN_{16}) (Ours)	25.96	17.09	12.52	9.91
Transformer (OP_8) (Ours)	21.83	13.85	10.34	8.28
Multi-feature fusion (CNN_8, CNN_{12}, CNN_{16}) (Ours)	29.34	19.86	14.81	11.83
Multi-feature fusion (CNN_8, CNN_{12}, CNN_{16}, OP_8) (Ours)	33.59	23.07	**17.25**	**13.75**

and passed onto the MFA module that now learns relations between the four features as shown in Equation (3).

$$\text{MFA} = (SA_{CNN8} \oplus SA_{CNN12} \oplus SA_{CNN_16} \oplus SA_{OP8}) \quad (3)$$

6.2. Decoder

The decoder begins to output predicted words upon receiving the start-of-sentence $< s >$ token. Only the words at the particular time-step are visible to the decoder. The embeddings are passed through a self-attention block for learning inter-relations between words. The decoder implements attention, feed-forward network, add and norm, and skip connections similar to the encoder. In addition to the above blocks, the decoder implements encoder-decoder joint attention in the multi-head attention block. The output obtained from the decoder is the spoken language equivalent of the sign language, thus performing continuous SLT.

7. Experiments and Results

7.1. Training Information

The multi-feature fusion architecture is trained on 7096 samples from the GSL dataset. We obtained the best performance by implementing 3 layers in the encoder and decoder, along with 2 heads for multi-head attention in both the encoder and decoder.

During training, ground truth words were fed in to the decoder at every time step and future words were masked out so that the model learning did not underfit. Whereas, during inference time, only predicted words at each time step are fed to the next time step, for a fair evaluation of the model. We selected an initial learning rate of $1e^{-04}$ and a weight decay rate of $1e^{-03}$. The model saturated after 80 epochs. We used an encoder embedding size of 256 and a custom decoder embedding size of $256 \times \#offeatures$.

We evaluated our models on the test set by calculating BLEU [30] scores. BLEU score is evaluated by comparing the predicted *n-grams* with the ground truth *n-grams*. BLEU 1 predicts *1-gram* words, i.e. it looks for matching individual words from predicted to ground truth. BLEU 1 scores are usually high due to this as the order of words does not matter. The task gets challenging as the model proceeds towards predicting longer grams. *2-gram* words compare two consecutive words, *3-gram* compare 3 consecutive words, and so on. Because we are performing continuous SLT, BLEU 3 and 4 metrics are the most crucial for our evaluation.

7.2. Performance Evaluation

To test the efficacy of our scaling mechanisms, we initially ran the single-input baseline transformer (on each of S_{CNN8}, S_{CNN12}, S_{CNN16}, S_{OP8}) using their respective individual features. From the results shown in Table 1 we can see that baseline scaled features perform better than some of the state-of-the-art methods like Conv2d-RNN, for BLEU 3 and 4. [3].

In addition to individual features, we perform a tri-feature experiment by passing only the three scaled CNN features through the multi-feature fusion architecture. Our model outperformed the best sequence model Conv2d-RNN, with attention [3] by achieving a 43.53% higher BLEU 4 score.

Our proposed model improves the translation results with the addition of OpenPose features and the 4-feature network outperforms the state-of-the-art [16] by achieving higher BLEU 3 and 4 scores.

8. Conclusion

In this work, we studied the effects of implicit boundary segmentation in signs and the effects of multi-feature fusion architecture for sign language translation without transcription. Using a sliding window approach we segmented the input video into multiple segments with overlapping frames. Our fusion architecture performed different levels of fusion, firstly it learned self-attention by learning relationships between different frames of the input video, and secondly, it performed multi-feature attention by learning the relations between the four sets of features.

Our model out-performed the recent state-of-the-art translation model by achieving higher BLEU 3 - BLEU 4 scores. It outperformed the state-of-the-art sequence networks with attention, by achieving a 43.53 % higher BLEU 4 score. Higher n-gram accuracy (BLEU 3, BLEU 4) indicates that the model is robust enough in predicting longer grams. In future, we will consider benchmarking on larger datasets.

9. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1846076.

10. References

[1] "Five parameters in sign language," https://www.lifeprint.com/asl101/pages-layout/parameters.htm/, [Online; accessed 24-March-2021].

[2] R. Pfau and J. Quer, "Nonmanuals: Their prosodic and grammatical roles," *Sign languages*, pp. 381–402, 2010.

[3] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in *CVPR*, 2018, pp. 7784–7793.

[4] "Glossing in sign language," https://www.lifeprint.com/asl101/topics/gloss.htm, [Online; accessed 23-April-2020].

[5] "Glossing in sign language," https://www.startasl.com/sign-language-symbols/, [Online; accessed 23-April-2020].

[6] O. Koller, N. C. Camgoz, H. Ney, and R. Bowden, "Weakly supervised learning with multi-stream cnn-lstm-hmms to discover sequential parallelism in sign language videos," *TPAMI*, vol. 42, no. 9, pp. 2306–2320, 2020.

[7] B. Mocialov, G. Turner, K. Lohan, and H. Hastie, "Towards continuous sign language recognition with deep learning," in *Proc. of the Workshop on the Creating Meaning With Robot Assistants: The Gap Left by Smart Devices*, 2017.

[8] O. Koller, H. Ney, and R. Bowden, "Deep learning of mouth shapes for sign language," in *ICCV*, 2015, pp. 85–91.

[9] O. Koller, H. Ney, and R. Bowden, "Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled," in *CVPR*, 2016, pp. 3793–3802.

[10] X. Li, C. Mao, S. Huang, and Z. Ye, "Chinese sign language recognition based on shs descriptor and encoder-decoder lstm model," in *Chinese Conference on Biometric Recognition*. Springer, 2017, pp. 719–728.

[11] L. Ding and A. M. Martinez, "Modelling and recognition of the linguistic components in american sign language," *Image and Vision Computing*, vol. 27, no. 12, pp. 1826–1844, 2009.

[12] H. Cooper, E.-J. Ong, N. Pugeault, and R. Bowden, "Sign language recognition using sub-units," *J. Mach. Learn. Res.*, vol. 13, no. 1, p. 2205–2231, Jul. 2012.

[13] D. Li, C. R. Opazo, X. Yu, and H. Li, "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison," *ArXiv*, vol. abs/1910.11006, 2019.

[14] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "Multi-channel transformers for multi-articulatory sign language translation," *arXiv preprint arXiv:2009.00299*, 2020.

[15] N. C. Camgoz, O. Koller, S. Hadfield and R. Bowden, "Sign language transformers: Joint end-to-end sign language recognition and translation," *arXiv preprint arXiv:2003.13830*, 2020.

[16] D. Li, C. Xu, X. Yu, K. Zhang, B. Swift, H. Suominen, and H. Li, "Tspnet: Hierarchical feature learning via temporal semantic pyramid for sign language translation," 2020.

[17] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[18] S. Venugopalan, M. Rohrbach, J. Donahue, R. J. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence - video to text," *CoRR*, vol. abs/1505.00487, 2015. [Online]. Available: http://arxiv.org/abs/1505.00487

[19] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[20] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[21] M. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *CoRR*, vol. abs/1508.04025, 2015. [Online]. Available: http://arxiv.org/abs/1508.04025

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NIPS*, vol. 30, pp. 5998–6008, 2017.

[23] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," *CoRR*, vol. abs/1901.02860, 2019. [Online]. Available: http://arxiv.org/abs/1901.02860

[24] A. Radford, "Improving language understanding by generative pre-training," 2018.

[25] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[26] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[27] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017.

[28] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, "Human pose estimation with iterative error feedback," 2016.

[29] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *CoRR*, vol. abs/1409.1259, 2014. [Online]. Available: http://arxiv.org/abs/1409.1259

[30] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 311–318. [Online]. Available: https://doi.org/10.3115/1073083.1073135