LLMs are Good Sign Language Translators (Supplementary Material)

Jia Gong[1†] Lin Geng Foo[1†] Yixuan He[1†] Hossein Rahmani[2] Jun Liu[1‡]
[1]Singapore University of Technology and Design [2]Lancaster University
{jia_gong,lingeng_foo,yixuan_he}@mymail.sutd.edu.sg,
h.rahmani@lancaster.ac.uk, jun_liu@sutd.edu.sg

To summarize our Supplementary Material: In Sec. 1 we report more experiment results of our method. Then, in Sec. 2 we provide more qualitative visualizations. Following that, we provide more implementation details in Sec. 3, and a summary of our codebook reconstruction algorithm in Sec. 4. Next, we provide the proof of Eq. 4 of the main paper in Sec. 5. Lastly, we discuss some possible future directions in Sec. 6.

1. More Experiments

Here, we report more experiment results on the Phoenix-2014T dev and test sets.

Impact of Context Prediction Pre-training. We conduct further investigations on VQ-Sign's context prediction pre-training by comparing against the following baseline: **Ours (w/ Autoencoding)** where we learn a discrete character-level codebook with a VQ-VAE [7] approach, i.e., via video self-reconstruction. Specifically, we follow [14] to conduct video autoencoding, and perform this autoencoding pre-training for approximately the same amount of time as our VQ-Sign's context prediction pre-training. As shown in Tab. 1, our proposed method significantly outperforms the baseline. This shows that pre-training the discrete character-level codebook via the context prediction task is more effective than autoencoding, which can be difficult for high-dimensional and complex videos.

Table 1. Ablation study of context prediction pre-training.

Method	Dev				Test			
	B1	B2	B3	B4	B1	B2	B3	B4
Ours (w/ Autoencoding)	42.84	32.41	25.63	21.00	41.27	30.73	24.17	19.80
Ours	46.88	36.59	29.91	25.25	45.21	34.78	28.05	23.40

Impact of Discrete Characteristics. To verify the impact of discrete characteristics, we further investigate the following baselines: 1) **Continuous Representation** where we feed the continuous outputs Z of the visual encoder E_v into the LLM without discretizing using the discrete character-level codebook. Note that, we train E_v in an end-to-end manner with $\mathcal{L}^{sim}$. 2) **Continuous Representation (w/ $\mathcal{L}^{cp}$)** where we additionally pre-train the visual encoder E_v via the context prediction loss $\mathcal{L}^{cp}$, and then fine-tune E_v via $\mathcal{L}^{sim}$. Furthermore, since these continuous baselines are unable to use the codebook reconstruction, for fair comparison, we further report results of a discrete baseline: **Discrete Representation (w/o Codebook Reconstruction)**. As shown in Tab. 2, the continuous baselines show significantly worse performance than the discrete settings, even when directly compared with the discrete baseline where codebook reconstruction is not used. These results show the efficacy of imparting discrete characteristics to the sign video representations.

Table 2. Ablation study of discrete characteristics.

Method	Dev				Test			
	B1	B2	B3	B4	B1	B2	B3	B4
Continuous Representation	26.95	17.04	12.11	9.41	25.63	16.10	11.20	8.42
Continuous Representation (w/ $\mathcal{L}^{cp}$)	31.13	21.36	15.84	12.47	31.17	21.42	15.62	12.19
Discrete Representation (w/o Codebook Reconstruction)	40.45	30.40	24.07	19.79	40.25	30.05	23.63	19.47
Ours	46.88	36.59	29.91	25.25	45.21	34.78	28.05	23.40

Impact of CRA's Design. Next, we verify the effectiveness of our CRA module's design. We compare against the following baselines: 1) **Ours w/o Pre-processing** where we do not pre-process our character-level token sequence to remove repeated tokens before performing the codebook reconstruction. 2) **Averaging Character-level Embeddings** where we combine character-level embeddings into word-level embeddings by taking the average of the character-level embeddings, instead of using the autoregressive model to fuse them. As shown in Tab. 3, our method outperforms all baselines, showing the efficacy of our designs.

Table 3. Ablation study of CRA's designs.

Method	Dev				Test			
	B1	B2	B3	B4	B1	B2	B3	B4
Ours w/o Pre-processing	44.15	33.98	27.46	22.97	43.30	32.85	26.22	21.66
Averaging Character-level Embeddings	42.56	32.23	25.76	21.32	42.28	31.95	25.34	20.96
Ours	46.88	36.59	29.91	25.25	45.21	34.78	28.05	23.40

Inference Speed. Moreover, we investigate the inference speed of our method on a set of videos whose average duration is around 3 seconds. We also report the inference speed of a variant: **Ours (w/ T5)** where we adopt T5 [8] (T5-3B-

† Equal contribution; ‡ Corresponding author

Example 1: Sign Word [$s_{213}s_{251}s_{213}$] <---> German Word [im Norden]

Ground truth: heute nacht gibt es im norden und nordosten viele wolken örtlich regnet oder nieselt es etwas (Tonight there will be a lot of clouds in the north and northeast, with some rain or drizzle in places) Sign sentence: s_{140}| s_{86}|$s_{213}s_{251}s_{213}$|s_{116}|s_{80}|$s_{213}s_{140}$| $s_{90}s_{213}$|s_{67}|s_{140} s_{116}|$s_{255}s_{213}$|$s_{116}s_{88}$
Ground truth: am sonntag im norden noch wolkig und vor allem in küstennähe einzelne schauer (On Sunday still cloudy in the north and a few showers, especially near the coast) Sign sentence: $s_{213}s_{164}$|$s_{213}s_{251}s_{213}$| $s_{90}s_{213}$|s_{119}|$s_{140}s_{213}$|$s_{116}s_{88}$
Ground truth: und morgen wird es dann in der südosthälfte nochmal ähnlich werden wie heute allerdings im nordwesten bereits dichtere wolken (And tomorrow in the southeast half it will be similar to today, although there will already be denser clouds in the northwest) Sign sentence: $s_{116}s_{59}$|s_{180}|$s_{116}s_{173}$|s_{80}|s_{213} s_{169}|$s_{197}s_{169}$|s_{140}|$s_{213}s_{251}s_{213}$|$s_{251}s_{140}$|$s_{156}s_{213}$

Example 2: Sign Word [$s_{150}s_{213}$\|$s_{116}s_{88}$] <---> German Word [recht Freundlich]

Ground truth: in der südosthälfte deutschlands bleibt es morgen unter leichtem hochdruckeinfluss noch recht freundlich (In the southeastern half of Germany it will remain quite friendly tomorrow with a slight influence of high pressure) Sign sentence: $s_{116}s_{91}$|s_{132}|$s_{140}s_{247}$|s_{140}|$s_{173}s_{213}$|$s_{16}s_{71}$|$s_{140}s_{213}$|$s_{150}s_{213}$|$s_{116}s_{88}$
Ground truth: richtung norden und westen ist es recht freundlich (Towards the north and west it is quite friendly) Sign sentence: $s_{116}s_{140}$|$s_{251}s_{213}$|$s_{150}s_{213}$|$s_{116}s_{88}$
Ground truth: am mittwoch in der nordhälfte regenschauer richtung süden ist es freundlicher (On Wednesday in the northern half of the rain showers towards the south it is friendlier) Sign sentence: s_{213}|s_{40}|s_{170}|s_{164}|s_{251}|$s_{140}s_{213}$|$s_{140}s_{213}$|$s_{150}s_{213}$|$s_{116}s_{88}$

Example 3: Sign Word [$s_{140}s_{204}s_{213}s_{116}s_{88}$] <---> German Word [sonnenschein]

Ground truth: sonst viel sonnenschein (otherwise lots of sunshine) Sign sentence: $s_{140}s_{247}$|$s_{213}s_{150}$|$s_{140}s_{204}s_{213}s_{116}s_{88}$
Ground truth: in der südwesthälfte morgen teils wolken teils längere zeit sonnenschein (in the southwest tomorrow partly cloudy partly sunshine for a longer time) Sign sentence: $s_{116}s_{140}$|$s_{173}s_{129}$|$s_{213}s_{140}$|$s_{90}s_{213}$|$s_{254}s_{213}$|$s_{140}s_{204}s_{213}s_{116}s_{88}$
Ground truth: und so erwartet uns eine mischung aus teilweise zähen nebelfeldern wolken und sonnenschein (And so we can expect a mixture of partly thick fields of fog, clouds and sunshine) Sign sentence: $s_{213}s_{205}$|$s_{247}s_{140}$|$s_{240}s_{205}s_{171}s_{207}$|$s_{213}s_{242}$| $s_{18}s_{90}$|s_{213}| $s_{140}s_{204}s_{213}s_{116}s_{88}$

Figure 1. Visualization of produced sign sentences. The character-level sign tokens are represented by s_i, and are composed into word-level sign tokens based on the groupings denoted by the separator '|'. We highlight the correlated spoken language words in red and the word-level tokens in blue. Overall, we find that the produced word-level sign tokens tend to correspond well with German words. Furthermore, our CRA module can find complicated word-level tokens that are a combination of quite a few character-level tokens, as shown in Example 3 above.

64bit) as our LLM instead of LLaMA [11] (LLaMA-7B-32bit). As illustrated in Tab. 4, our method is faster and has better accuracy than the variant using T5. The smaller size yet slower inference speed of T5-3B-64bit compared to LLaMA-7B-32bit can be attributed to the structural differences between these two LLMs. Specifically, T5-3B-64bit follows an encoder-decoder architecture, while LLaMA-7B-32bit is a decoder-only model. Therefore, T5-3B-64bit tends to require more time for inference due to its additional processing steps during encoding. Overall, our approach maintains rapid inference capabilities relative to video length, while showing significant performance gains compared to existing methods (as shown in Tab. 1 and Tab. 2 of main paper).

Results on Larger Dataset. Furthermore, we present results on the large How2Sign dataset [1]. Specifically, we follow the method outlined by [10], utilizing a pre-trained Inflated 3D Convolutional Neural Network (I3D) as initialization for our visual encoder. On the How2Sign dataset, our approach achieves a slight improvement over the previous state-of-the-art [10] (10.32 BLEU4 vs. 8.03 BLEU4).

Table 4. Comparison of inference speed.

Method	B4	SignLLM time (s)	LLM time (s)	Total time (s)
Ours (w/ T5)	22.51	0.53	4.94	5.47
Ours (w/ LLaMA)	23.40	0.53	1.92	2.45

2. More Visualization

More Visualization of Sign Sentences. In Fig. 2 of the main paper, we presented some visualizations of produced sign sentences, here we present more of such vi-

Sign Word [$s_{140}s_{204}$] <---> German Word [sonne]

Ground truth: sonst ein wechsel aus sonne und wolken
(otherwise an alternation of sun and clouds)
Sign sentence: $s_{116}|s_{140}s_{247}|s_{213}s_{241}|s_{213}|s_{140}s_{204}|s_{140}s_{90}s_{213}|s_{116}s_{88}$

Ground truth: auch am montag neben sonne zum teil schwere gewitter
(also on monday besides sun partly heavy thunderstorms)
Sign sentence: $s_{116}s_{170}s_{54}|s_{169}s_{132}s_{169}|s_{140}s_{204}|s_{213}s_{242}|s_{140}s_{3}|s_{213}|s_{242}s_{213}|s_{116}s_{88}|$

Sign Word [$s_{140}s_{173}s_{213}$] <---> German Word [im süden]

Ground truth: besonders im südlichen bergland einzelne wärmegewitter
(Individual warm thunderstorms, especially in the southern mountains)
Sign sentence: $s_{86}s_{247}s_{240}|s_{213}|s_{140}s_{173}s_{213}|s_{31}s_{71}|s_{140}s_{213}|s_{116}s_{88}$

Ground truth: im süden entladen sich später hier und da heftige gewitter
(In the south, violent thunderstorms erupt here and there later)
Sign sentence: $s_{140}s_{173}s_{213}|s_{91}s_{213}|s_{89}s_{213}|s_{140}\ s_{213}|s_{140}|s_{242}s_{213}|s_{116}s_{88}$

Figure 2. Visualization of sign tokens' hierarchical structure. The word-level sign tokens of interest are highlighted in blue and the corresponding German words are highlighted in red. (Top) We observe a correspondence between a word-level sign token composed from the character-level token sequence $[s_{140}s_{204}]$, and the German word $[sonne]$. (Bottom) We observe that, by replacing the character-level token s_{204} with $s_{173}s_{213}$ for the word-level token $[s_{140}s_{204}]$, we can get a new word-level sign token $[s_{140}s_{173}s_{213}]$ with a different meaning that corresponds with the German word $[im\ süden]$.

sualizations. In Fig. 1, we present visualizations of sign sentences (comprised of word-level sign tokens) alongside their corresponding spoken language sentences. The ground truth German sentences are provided, as well as the corresponding English translations (to facilitate understanding for readers who do not speak German). Overall, we observe significant correlations between the word-level sign tokens produced by our CRA module and spoken language words. For example, in the first example, the word-level sign token $[s_{213}s_{251}s_{213}]$ consistently corresponds to the German word sequence $[im\ norden]$, while in the second example, the word-level sign tokens $[s_{150}s_{213}]$ and $[s_{116}s_{88}]$ consistently align with the German word sequence $[recht\ Freundlich]$. Moreover, in the third example, we observe that our codebook reconstruction algorithm can find longer sequences of character-level tokens, e.g., $[s_{140}s_{204}s_{213}s_{116}s_{88}]$ that corresponds to a specific German word $[sonnenshein]$. These visualizations suggest that our word-level sign tokens contain semantic meaning and also show the efficacy of our optimal transport-based codebook reconstruction algorithm.

Visualization of Sign Tokens' Hierarchical Structure. Next, we aim to visualize the hierarchical structure of the produced sign tokens. In Fig. 2 we present visualizations of sign sentences (composed of word-level sign tokens) alongside their corresponding spoken language sentences. At the top of Fig. 2, we observe that the combination of two character-level sign tokens ($[s_{140}s_{204}]$) corresponds to the German word ($[sonne]$). Next, at the bottom of Fig. 2, we also observe that if the second character-level token

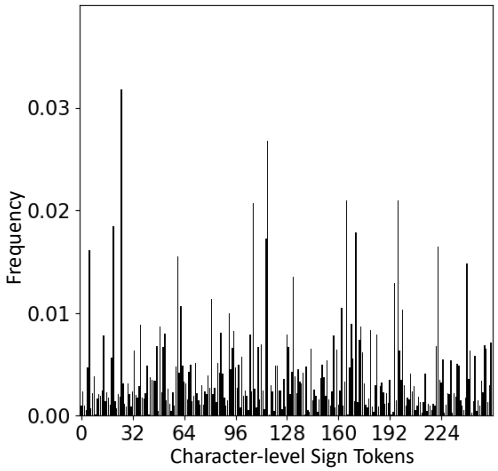

Figure 3. Visualization of the usage frequency of the character-level sign tokens in the character-level codebook.

is changed, i.e., modifying the sequence from $[s_{140}s_{204}]$ to $[s_{140}s_{173}s_{213}]$, then the sequence gets mapped to another word-level sign token. Specifically, after the modification, the resulting word-level token gets mapped to another German word ($[im\ süden]$) with a different meaning. This suggests that individual character-level sign tokens (which may not contain much semantic meaning on their own) are assembled to form word-level sign tokens that are associated with specific semantic meaning. This observed character-to-word relationship mirrors the hierarchical structure present in spoken languages.

Another example of hierarchical structure can be observed between the top example of Fig. 2 with the German Word $[sonne]$ (i.e., "sun") and the third example of Fig. 1 with the German Word $[sonnenschein]$ (i.e., "sunshine"). Specifically, the character-level token sequence $[s_{140}s_{204}]$ corresponds to the word $[sonne]$/sun, while appending a few character-level tokens to the sequence (to obtain $[s_{140}s_{204}s_{213}s_{116}s_{88}]$) makes it correspond to the word $[sonnenschein]$/sunshine. This observed relationship between these two words also mirrors the hierarchical structure of spoken languages.

Visualization of Character-level Token Usage Frequency. We calculate the frequency of each character-level token in the training set and present their usage frequency (as a percentage of all retrieved character-level tokens) in Fig. 3. We observe that all character-level tokens have been used and some tokens are used much more frequently than others. This pattern aligns with the nature of spoken language, where some characters are commonly used, and others are used less often.

3. More Implementation Details

Here, we describe the prompting strategy we use to guide the large language model (LLM) to focus on the sign language translation (SLT) task and avoid open-ended generation. Firstly, to instruct the LLM to perform the translation task, our prompt includes a concise task instruction, e.g, "Translate the given sentence into German:". Note that, during the training stage, we further enhance prompt diversity by paraphrasing and translating the above task instruction into various other prompts, with the aid of the LLM. Moreover, we employ LLaMA [11] as our LLM, which tends to generate open-ended content that are unrelated to the translation task. Thus, to handle this issue, we further *include multilingual translation pairs* in our prompt as in-context examples, which help to regulate the text generation to keep to the translation task when generating output text. More precisely, at the *start of training*, we sample 20 sentences from the training set and translate them to other languages (e.g, English, Japanese and Russian) via a translation engine (e.g., Google Translate), where each of these sentences and their translations form a paired translation sample. Then, in each iteration, we randomly *select one* of these 20 paired translation samples and append it to the task instruction in the following prompt format: "Translate the given sentence into German. The input and response pair is as follows: Input: [Input], Response: [Response]". In the above prompt, the provided input is the foreign language sentence (e.g., English), and the response is the translation in the desired language (e.g., German for Phoenix2014T dataset). Note that, at test time, we also sample one paired translation sample from the 20 training set sentences to assist LLaMA to handle the SLT task. By providing these multilingual translation examples to LLaMA, we can encourage LLaMA to generate output text that are direct translations of the input sign sentence, without generating superfluous open-ended content that are unrelated to the translation task.

Next, we provide more details of our codebook reconstruction. Firstly, to set-up the distance matrix $\boldsymbol{D}$, we need to identify $r \times m$ word candidates. To achieve this, we apply [9] to find the top $r \times m$ most frequent character sequences (following [13]). We also remark that, we do not constrain the number of characters that each word-level token can contain, although we observe that word-level tokens often consist of 2-6 characters. Note that, for most candidate words, each of them contains characters that are all different from each other (i.e., without duplicates within the candidate words), which is facilitated by our pre-processing which removes consecutively repeated tokens.

For the Sign-Text Alignment Loss in Sec. 3.3 of the main paper, we apply two MMD losses to narrow the overall sign-text gap – one for the character-level tokens and one for word-level tokens. This aligns both character-level tokens and word-level tokens with the LLM's text embedding space, facilitating better understanding of them by LLMs. Specifically, each pre-trained LLM has a tokenizer that maps each word to an embedding, i.e., a q token in Eq. 6 of main paper. Therefore, by feeding all text in the training set to tokenizer, we get the set of q tokens. Then, our MMD loss (in EQ. 6 of main paper) minimizes *gap between distributions* of the set of p tokens and the set of q tokens, where the p tokens can be the character-level tokens or the word-level tokens. In other words, our MMD loss does not rely on direct correspondence between each specific p and q token.

Here, we describe how we generate an embedding for each word-level sign token based on its character-level sign tokens. Specifically, to generate an embedding for the word-level sign tokens, we employ a recurrent network whose structure is the same as the recurrent module g in VQ-Sign module, i.e., one Convolutional Gated Recurrent Layer with a kernel size of (1, 1). Given a small set of character-level tokens that combine to form a word-level token, the network recurrently processes the character-level tokens' embeddings, to ultimately produce an embedding for the word-level sign token. Moreover, to streamline the training process, we initialize the weights of this recurrent network with the pre-trained g from VQ-Sign. Thus, the final recurrent network is a fine-tuned version of g.

Overall, we conduct the whole training process on 4 NVIDIA RTX A5000 GPU cards within 60 hours.

4. Summary of Word-level Codebook Reconstruction Algorithm

In Sec. 3.3 of the main paper, we presented the word-level codebook reconstruction algorithm as part of our CRA module. Here, we further include a summary of our word-level codebook reconstruction algorithm in Algorithm 1.

Algorithm 1: Word-level Codebook Reconstruction Algorithm

Input: Character-level codebook $\mathbb{S}^c$, increment m, initial word-level codebook $S_0^w = \mathbb{S}^c$
$Expansion = True$, initial step $r = 0$, codebook candidates = $[S_0^w]$
while *Expansion* **do**
  $r = r + 1$
  Expand codebook size to $r \times m$
  Construct the r^{th} word-level codebook $\mathbb{S}_r^w$ from $\mathbb{S}^c$ through optimal transport formulation
  Append $\mathbb{S}_r^w$ to codebook candidates
  Calculate the entropy decrease: $\Delta_r = \mathcal{H}_{\mathbb{S}_r^w} - \mathcal{H}_{\mathbb{S}_{r-1}^w}$
  if $\Delta_r > \Delta_{r-1}$ **then**
    $Expansion = False$
Output $\mathbb{S}_{r-1}^w$ from codebook candidates.

5. Proof of Eq. 4 in Main Paper

In this section, we provide the proof of Eq. 4 in the main paper. The proof (which follows [13]) starts from the definition of entropy (Eq. 3 of the main paper), and is as follows:

$$\mathcal{H}_{\mathbb{S}_r^w} = -\sum_{w_j \in \mathbb{S}_r^w} P(w_j) \log P(w_j) \tag{1}$$

$$= -\sum_{w_j \in \mathbb{S}_r^w} \sum_{s_i \in \mathbb{S}^c} P(w_j, s_i) \log P(w_j) \tag{2}$$

$$= -\sum_{w_j \in \mathbb{S}_r^w} \sum_{s_i \in \mathbb{S}^c} P(w_j, s_i) \log \left[P(w_j, s_i) \cdot \frac{P(w_j)}{P(w_j, s_i)} \right] \tag{3}$$

$$= -\sum_{w_j \in \mathbb{S}_r^w} \sum_{s_i \in \mathbb{S}^c} P(w_j, s_i) \log P(w_j, s_i) - \sum_{w_j \in \mathbb{S}_r^w} \sum_{s_i \in \mathbb{S}^c} P(w_j, s_i) \log \frac{P(w_j)}{P(w_j, s_i)} \tag{4}$$

$$= -\sum_{w_j \in \mathbb{S}_r^w} \sum_{s_i \in \mathbb{S}^c} P(w_j, s_i) \log P(w_j, s_i) - \sum_{w_j \in \mathbb{S}_r^w} \sum_{s_i \in \mathbb{S}^c} P(w_j, s_i)(-\log \frac{P(w_j, s_i)}{P(w_j)}) \tag{5}$$

$$= -\sum_{w_j \in \mathbb{S}_r^w} \sum_{s_i \in \mathbb{S}^c} P(w_j, s_i) \log P(w_j, s_i) - \sum_{w_j \in \mathbb{S}_r^w} \sum_{s_i \in \mathbb{S}^c} P(w_j, s_i)(-\log P(s_i|w_j)). \tag{6}$$

6. Future Work

In this work, inspired by the impressive text generation capabilities of LLMs [3, 17], we explore leveraging off-the-shelf LLMs for SLT, by imparting *discrete* and *hierarchical* characteristics to the sign video representations. Future works involve exploring other self-supervised learning strategies [5, 6] for learning discrete representations, or other ways of incorporating hierarchical structure, e.g., grammar induction [12]. Other directions include exploring such an approach for related tasks such as video captioning [4, 15] and video understanding [2, 16].

References

[1] Amanda Duarte, Shruti Palaskar, Lucas Ventura, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier Giro-i Nieto. How2sign: a large-scale multimodal dataset for continuous american sign language. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2735–2744, 2021. 2

[2] Lin Geng Foo, Jia Gong, Zhipeng Fan, and Jun Liu. System-status-aware adaptive network for online streaming video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10514–10523, 2023. 5

[3] Lin Geng Foo, Hossein Rahmani, and Jun Liu. Ai-generated content (aigc) for various data modalities: A survey. *arXiv preprint arXiv:2308.14177*, 2, 2023. 5

[4] Xin Gu, Guang Chen, Yufei Wang, Libo Zhang, Tiejian Luo, and Longyin Wen. Text with knowledge graph augmented transformer for video captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18941–18951, 2023. 5

[5] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 5

[6] Tianjiao Li, Lin Geng Foo, Ping Hu, Xindi Shang, Hossein Rahmani, Zehuan Yuan, and Jun Liu. Token boosting for robust self-supervised visual transformer pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24027–24038, 2023. 5

[7] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 1

[8] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 1

[9] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015. 4

[10] Laia Tarrés, Gerard I Gállego, Amanda Duarte, Jordi Torres, and Xavier Giró-i Nieto. Sign language translation from instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5624–5634, 2023. 2

[11] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2, 4

[12] Bo Wan, Wenjuan Han, Zilong Zheng, and Tinne Tuytelaars. Unsupervised vision-language grammar induction with shared structure modeling. In *International Conference on Learning Representations*, 2022. 5

[13] Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. Vocabulary learning via optimal transport for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7361–7373, 2021. 4, 5

[14] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021. 1

[15] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and

Cordelia Schmid. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10714–10726, 2023. 5

[16] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 543–553, 2023. 5

[17] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023. 5