

Subobject-level Image Tokenization

Delong Chen¹ Samuel Cahyawijaya¹ Jianfeng Liu² Baoyuan Wang² Pascale Fung¹
¹Hong Kong University of Science and Technology ²Xiaobing.AI

Abstract

Transformer-based vision models typically tokenize images into fixed-size square patches as input units, which lacks the adaptability to image content and overlooks the inherent pixel grouping structure. Inspired by the *subword* tokenization widely adopted in language models, we propose an image tokenizer at a *subobject* level, where the subobjects are represented by semantically meaningful image segments obtained by segmentation models (e.g., segment anything models). To implement a learning system based on subobject tokenization, we first introduced a Sequence-to-sequence AutoEncoder (SeqAE) to compress subobject segments of varying sizes and shapes into compact embedding vectors, then fed the subobject embedding vectors into a large language model for vision language learning. Empirical results demonstrated that our subobject-level tokenization significantly facilitates efficient learning of translating images into object and attribute descriptions compared to the traditional patch-level tokenization. Codes and models will be open-sourced at <https://github.com/ChenDelong1999/subobjects>.

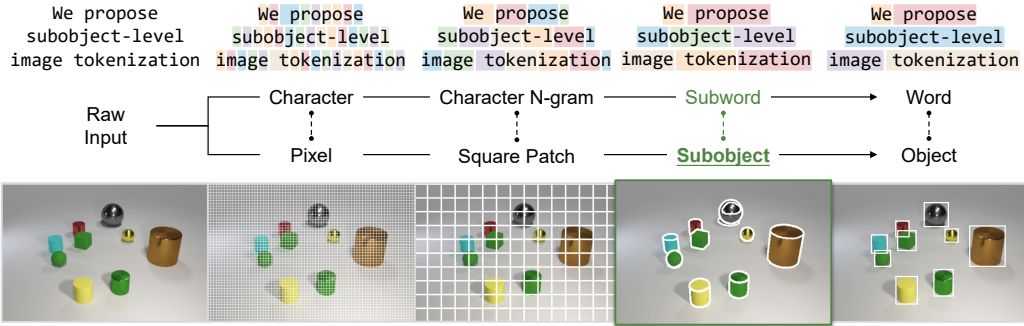


Figure 1: The connection between tokenization methodologies in language modelling and image modelling. Our proposed subobject-level image tokenization is corresponded to the subword-level textual tokenization, which has been proven to be superior compared to other alternatives.

1 Introduction

In the field of Natural Language Processing (NLP), researchers have extensively investigated a variety of tokenization approaches over years [1], but have converged to the subword-level tokenization recently, represented by Bytepair Encoding (BPE) [2], Unigram [3], and the SentencePiece tokenizer [4] adopted by many language models. Compared to the character-level and word-level counterparts, subword-level tokenization is better at capturing the morphology of a word which not only avoids the explosion of vocabulary size but also facilitates compositional generalization, e.g., the word “tokenization” and “generalization” can be decomposed into two tokens each, yielding “token” and “general” as the prefix, while sharing the same suffix token “-ization”. Additionally, subword-level

Work in progress.



Figure 2: The initial results from the “*segment everything*” mode of Segment Anything Model (SAM) [8] leaves many pixels in blank. To avoid information loss, we perform post-processing of mask expansion and gap infilling to ensure the subobject segmentation covers all pixels.

tokenization brings out-of-vocabulary generalization, minimizing the production of unknown tokens during the tokenization process.

In Computer Vision (CV), since our visual world has not been discretized and optimized for communication as a language, it is more sparse, redundant, and noisy than textual data. Directly using raw pixels as input units (*e.g.*, pixel-level tokenization adopted by ImageGPT [5]) leads to an excessively large number of tokens and unnecessary modeling of low-level relationships between neighboring pixels. The Vision Transformer (ViT) architecture [6], which is the dominant type of vision model in the current field, raises the tokenization level from pixels to square patches. However, as demonstrated in Fig. 1, such patch-level image tokenization corresponds to a character N-gram-level textual tokenization method, which tends to be both ineffective and inefficient due to the ignorance of semantic boundaries and humongous vocabulary size [7, 2]. Similar to N-gram tokenization, the patch partitioning operation is not adaptive to the morphology of objects, ignoring the inherent pixel-grouping structure in the image.

In this paper, inspired by the performant subword-level text tokenization, we introduce the concept of “*subobject*”-level image tokenization, which lies in an intermediate level between objects and pixels, akin to subwords between words and characters. Subobjects are visual entities (*e.g.*, parts of objects) with perceptually meaningful visual structures. It can be obtained through image segmentation, such as using Segment Anything Models (SAM) [8]. The concept of subobject is related to the concept of *superpixels* [9, 10] for low-level vision and the *part segmentation* [11] task in high-level vision, but it emphasizes the requirements of being semantically meaningful, open-vocabulary, and panoptic [12], also highlights the application to image tokenization and the connection to subword in NLP.

At the methodology level, this paper presents two types of neural architectures for effectively creating a learning system based on subobject-level image tokenization. **Firstly**, we propose a Sequence-to-sequence AutoEncoder (SeqAE) to compress subobject segments of varying sizes and shapes into compact embedding vectors. Compared to downsampling and fitting irregular segments into a square input window, SeqAE is able to reserve more information when handling segments with extreme aspect ratios. **Secondly**, we designed a simple yet effective architecture of Large Vision Language Model (LVLM), which incorporates subobject tokens into a Large Language Model (LLM) by treating them as textual subword tokens in new languages [13].

Empirically, we trained a SeqAE on the SA-1B dataset [8], then trained an LVLM with subobject-level image tokenization based on the Phi-2 model [14], using a synthetic captioning dataset created from CLEVR [15]. Our results demonstrate that subobject-level tokenization enables significantly accelerated vision-language learning compared to the standard ViT-style or Fuyu-style [16] patch-level tokenization, while at the same time achieving higher accuracies in counting objects and recognizing visual attributes such as size, material, and shape by a large margin.

2 Method

In this section, we introduce our method of creating a learning system based on subobject-level tokenization. Our method consists of the following three steps: *segmentation*—obtaining subobject boundaries from images (section 2.1); *embedding*—converting raw pixels of subobject into compact vector embedding (section 2.2); *modeling*—building a model that takes embedded subobject tokens as inputs for vision-language learning (section 2.3).

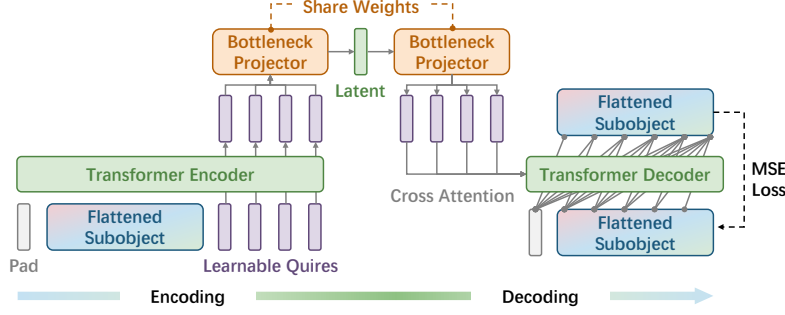


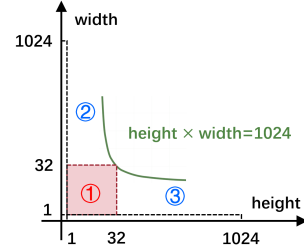
Figure 3: The architecture of our proposed Sequence-to-sequence AutoEncoder (SeqAE) for subobject embedding. SeqAE extracts compact latent variables from an image segment using learnable queries vectors and autoregressively decode the segment from the latent variables through the cross-attention mechanism.

2.1 Segment Everything into Subobjects

The desired subobject boundaries should be semantic meaningful, open-vocabulary, and comprehensive. Among other alternatives such as superpixel segmentation, semantic/instance/panoptic segmentation, the “*segment everything*” mode [17, 18] (also known as automated mask generation) of the Segment Anything Model (SAM) [8] has more advantages on satisfying the requirements of being semantic meaningful and open-vocabulary. However, the results of “*segment everything*” are not guaranteed to be comprehensive. As shown in Fig. 2, there could be many pixels (shown in white) that are not covered by any mask in the “*segment everything*” results. These uncovered pixels usually correspond to the background or tiny gaps between neighboring segments. To ensure comprehensiveness, we conduct post-processing by applying convolution on the segmentation mask with a small kernel to expand the masks and filling the gaps (similar to binarizing blurred masks), then we do connected component labeling on pixels that are still not covered by any segments. Segments post-processed by such mask expansion and background filling can cover all pixels in the image and avoid any information loss.

2.2 SeqAE for Subobject Embedding

Subobject segments have irregular sizes and shapes. Although it’s possible to fit segments into square perception windows by padding to the longest side, it would be very inefficient when facing segments with extreme aspect ratios. Consider a Transformer encoder with a square perception window of 32×32 pixels (*i.e.*, 1024 pixels), it could only losslessly encode segments within the ① area on the right. However, as shown on the right, with same budget of 1024 context length, it is possible to encode wider ② and higher ③ segments under the green curve (inverse proportion function) without any downsampling operations.



We introduce Sequence-to-sequence AutoEncoder (SeqAE) to address this issue. In SeqAE, raw segment pixels and masks are *flattened* into data sequences to make full use of the context length. SeqAE is trained to compress subobject segments into compact embeddings via self-supervised autoencoding objective. As shown in Fig. 3, the encoder extracts a latent vector from the input data sequence, then the decoder reconstructs the inputs autoregressively. SeqAE shares many architectural similarities with the vanilla encoder-decoder Transformer language model for neural machine translation [19], but has the following two key modifications:

Real-valued regression instead of categorical prediction: In language models, the decoder predicts the categorical probabilistic distribution of the next token and compares it with ground truth discretized one-hot embedding via cross-entropy. For pixels, although it is possible to regard each RGB pixel as a 256^3 -way or three 256-way categorical distribution, it either results in an extremely large vocabulary size or increases the number of tokens by three times. More importantly, it ignores the inherent continual nature of pixel intensity and loses the relational information between pixel intensity values

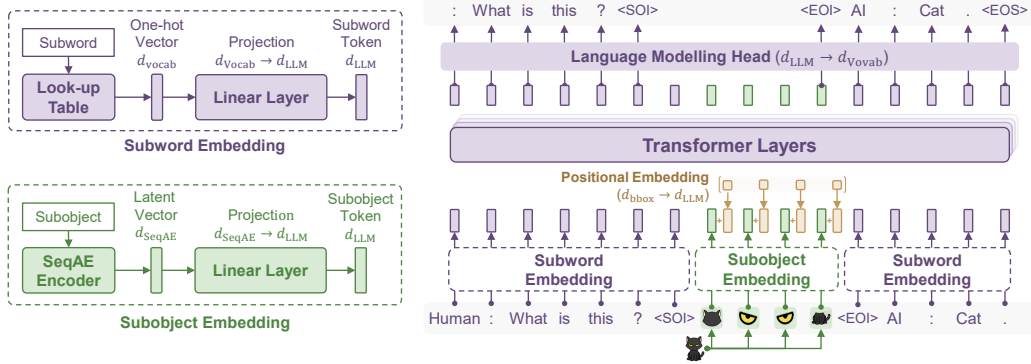


Figure 4: LLM to LVLM adaptation using our subobject-level image tokenization. We project the latent vector from SeqAE using a learnable linear weight and feed them into the LLM as parts of the input. We add a two-dimensional positional embedding to each subobject token to provide the position information of each subobject in the image.

(e.g., the model does not know that red is closer to purple but much further to green). Therefore, in SeqAE we directly use the normalized real-valued pixel intensity as the data sequence, and then applies mean squared error (MSE) regression loss on the decoder output.

Extracting compact latents via learnable queries and bottleneck projector: Inspired by Perceiver Resampler [20], learnable query tokens are appended to the input data sequence. They interact with pixel tokens and integrate their information after going through the encoder layers. We further add a linear layer on top of these query tokens in the encoder’s last layer to reduce the dimension from $d_{model} \times n_{query}$ to d_{SeqAE} , and use the same layer (transposed) to reconstruct the query tokens, which the decoder can cross-attend to. This linear layer acts as a bottleneck to encourage information compression.

2.3 LVLM based on Subobject-level Image Tokenization

Our methodology of inserting subobject tokens into LLMs is simple and straightforward. Inspired by Wang et al. [13], we treat them as textual subword tokens in new languages, and then creating LVLM from LLM becomes equivalent to adding a foreign language to an LLM. As shown on the left side of Fig. 4, the process of obtaining Transformer’s input tokens from subword and subobject are conceptually similar. On the right side of Fig. 4, the subobject tokens are interleaved with subword tokens at the same level, only with a pair of new special tokens <SOI> and <EOI> marking the start and the end of subobject tokens from a single image.

However, considered as a foreign language, the image has one fundamental difference compared to natural languages, which is its *dimensionality*. To accommodate this unique nature, we make the following two technical modifications:

Additional positional embedding for subobject tokens: Since the original positional embedding existing in the LLM can only represent one-dimensional order relationships, we introduce additional two-dimensional positional embedding for subobject tokens. We use the absolute bounding box coordinates (in $[x, y, w, h]$ format) of the segmentation mask to represent the position of subobjects. As shown in Fig. 4, we train a linear layer to project the bounding box into the same dimension of subobject tokens, then add them together and feed the result into Transformer layers.

No autoregressive prediction for subobject tokens: Images are two-dimensional projections of our three-dimensional visual world. These subobjects do not form any one-dimensional causal structure similar to natural language, making “next subobject token prediction” irrelevant. Therefore, during the LVLM training, we only calculate the cross-entropy loss on textual subword tokens while skipping the subobject tokens.

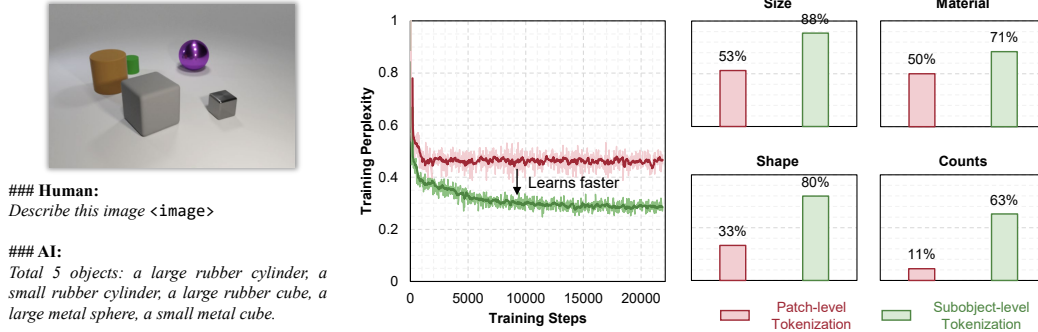


Figure 5: **Left:** An example of our synthesized image caption from CLEVR [15] dataset. <image> indicates the position of subobject tokens. **Middle:** With subobject-level image tokenization, vision-language modelling becomes significantly faster. **Right:** The accuracy of generated descriptions in terms of size, material, shape, and count. Subobject-level method outperform standard patch-level baselin by a large margin.

3 Experiment

3.1 Implementation of SeqAE

Model. Both the encoder and the decoder of SeqAE have 16 layers, where each layer is a standard Transformer layer with $d_{\text{model}} = 768$, $d_{\text{FFN}} = 4096$, and 12 attention heads. We use 16 learnable queries, each of them is a vector with a dimension of $d_{\text{model}} = 768$. The encoder output d_{SeqAE} is set to 768, and therefore the bottleneck projector is a linear layer of $16 \times 768 \rightarrow 768$. We use a context length of 1024 tokens, which can losslessly accommodate segments with width \times heights ≤ 1024 (e.g., 32×32 , 64×16 , 16×64 segments, etc). Larger segments are down-sampled to 1024 tokens. The entire SeqAE model with the above configurations has a total of 327 million parameters.

Data. We use the SA-1B dataset [8] to train the SeqAE model since it contains a larger-scale (1 billion) high-quality segmentation masks in of million images from various visual domains.

Training. We train SeqAE on the SA-1B dataset with a single-node $8 \times$ NVIDIA A100 (80GB) server. We use a batch size of 16 per GPU and a learning rate of $1e-5$.

3.2 Implementation of LVLM

Model. We use the Phi-2 model¹, which is a base LLM with 2.7B parameters trained on high-quality textual data [14]. We apply LoRA [21] on its query/key/value projectors and FFN layers. Two trainable linear layers are newly initialized to project subobject embedding from the frozen SeqAE encoder and the bounding box coordinates to the input token space of the LLM.

Data. We create a synthetic image captioning dataset from CLEVR [15], which contains scene graph annotations giving ground-truth locations, attributes, and relationships for objects. As shown in Fig. 5 left, we convert the scene graph annotations into textual descriptions of object counts, size, material, and shape. We arrange objects according to their positions (from left to right). We use 70k samples for training and evaluate 2k unseen samples.

Training. We use an effective batch size of 32 to train the model for 10 epochs. We use a cosine learning rate scheduler with a starting learning rate of $1e-4$. MobileSAM-v2 [18] is used for subobject segmentation. Subobject segments and embeddings are cached in advance to boost training efficiency.

3.3 Results

We compare the LVLM based on the subobject-level image tokenizer to a patch-level baseline, which divides the input image into 32×32 square patches – one of the most frequently used patch sizes in ViT models. Same as ViT, the patch token embedding is a linear layer that projects flattened

¹<https://huggingface.co/microsoft/phi-2>

patches ($32 \times 32 \times 3$) to the same dimension of SeqAE patents, and is also trained with autoencoding reconstruction objective. Other settings of this baseline are the same as the subobject-based LVLM.

The experimental results are presented in Fig. 5. On the left side, it shows that subobject-level tokenization enables a significantly faster decrease in training perplexity, showing that the same model can learn much faster when replacing patch-level tokenization with a subobject level one. On the right side, we present the evaluation of model-generated captions on 2k unseen testing images. We parse the generated captions (generated captions are 100% parseable), and respectively calculate the prediction accuracy of object size, material, shape, and counts with the ground truth. The results show that the model based on subobject-level tokenization outperforms the baseline by a large margin in all of the four aspects.

4 Conclusion

In this paper, we introduce subobject-level image tokenization which is a viable alternative to patch-level tokenization for vision-language learning. Our preliminary results demonstrate that compared to the standard patch-level baseline, subobject-level tokenization can accelerate vision-language learning, achieve higher accuracy in counting objects, and recognize visual attributes.

References

- [1] Sabrina J. Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y. Lee, Benoît Sagot, and Samson Tan. Between words and characters: A brief history of open-vocabulary modeling and tokenization in NLP. *CoRR*, abs/2112.10508, 2021.
- [2] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016.
- [3] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 66–75. Association for Computational Linguistics, 2018.
- [4] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71. Association for Computational Linguistics, 2018.
- [5] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, 2020.
- [6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.
- [7] Paul McNamee and James Mayfield. Character n-gram tokenization for european language text retrieval. *Inf. Retr. Boston.*, 7(1/2):73–97, January 2004.
- [8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 3992–4003. IEEE, 2023.
- [9] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*, pages 10–17. IEEE Computer Society, 2003.

- [10] David Stutz, Alexander Hermans, and Bastian Leibe. Superpixels: An evaluation of the state-of-the-art. *Comput. Vis. Image Underst.*, 166:1–27, 2018.
- [11] Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jieneng Chen, Shuai Liu, Cheng Yang, Qihang Yu, and Alan L. Yuille. Partimagenet: A large, high-quality dataset of parts. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part VIII*, volume 13668 of *Lecture Notes in Computer Science*, pages 128–145. Springer, 2022.
- [12] Alexander Kirillov, Kaiming He, Ross B. Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 9404–9413. Computer Vision Foundation / IEEE, 2019.
- [13] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: BEIT pretraining for vision and vision-language tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 19175–19186. IEEE, 2023.
- [14] Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need II: phi-1.5 technical report. *CoRR*, abs/2309.05463, 2023.
- [15] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1988–1997. IEEE Computer Society, 2017.
- [16] Adept Team. Fuyu-8B: A Multimodal Architecture for AI Agents. <https://www.adept.ai/blog/fuyu-8b>, 2023. [Accessed 19-02-2024].
- [17] Chaoning Zhang, Dongsheng Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight SAM for mobile applications. *CoRR*, abs/2306.14289, 2023.
- [18] Chaoning Zhang, Dongshen Han, Sheng Zheng, Jinwoo Choi, Tae-Ho Kim, and Choong Seon Hong. Mobilesamv2: Faster segment anything to everything. *CoRR*, abs/2312.09579, 2023.
- [19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.
- [20] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.
- [21] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.