ECE 1512 Digital Image Processing and Applications 2024 Fall Project B

Linfeng Ye, and Samuel Bernard
Edward S. Rogers Sr. Department of Electrical & Computer Engineering
University of Toronto

Abstract

In the initial section of this report, we conduct a thorough examination of the VILA model and its extensions. Similarly to the report on Mamba, the comprehensive study will review the core characteristics of VILA, identify significant technical innovations, and potential areas of improvement. As such, our goal is to provide a detailed summary of the VILA model's contributions and discuss how this model outperforms state-of-the-art Visual Language Models (VLMs) such as LLaVA-1.5.

In this second part of the report, we analyze an efficiency bottleneck inherent to VLMs and the VILA model architecture, and propose a novel approach designed to mitigate this limitation. Specifically, the time complexity introduced by the self-attention mechanism in the ViT architecture, which forms the basis of Transformer encoder layers in VLMs, scales quadratically with input length. To mitigate this, we propose the implementation of a Dynamic Depth Visual Encoding model aimed at reducing this complexity. To validate the performance of this model, we compare it against a baseline visual encoding model using the MNIST dataset. Both the testing accuracy and floating-point operations per second are recorded. These findings and improvements could serve as inspiration for further work on VILA, and provide a foundation for developing more efficient and robust Visual Language Models in the future. The code for this improvement analysis is publicly available at https://github.com/Linfeng-Ye/ECE1502_projectB

Index Terms

Deep learning, VILA, Pre-training, Large language model, Visual language model, Vision language model, In-context learning, Dynamic depth visual encoding

I. Review of VILA

A. Basic Concepts

1) Explain in detail the methodologies of VILA:

Large language models (LLMs) have recently gained attention for completing natural language tasks due to their remarkable properties. As such, these models have also become appealing for visual tasks, as integrating them would allow computer vision (CV) and natural language processing (NLP) to learn to associate visual features with linguistic expressions. Visual Language Models (VLMs) function by capturing spatial features from images, encoding textual information, mapping data from both modalities, and using Vision Transformers (ViTs) and other preprocessing techniques to connect visual and linguistic elements. This allows the model to perform image captioning, visual question answering, and image-text matching. However, integrating vision foundation models with LLMs presents a significant challenge, as both are typically pre-trained separately before undergoing joint vision-language training. This sequential training process can lead to suboptimal alignment between visual and linguistic components. Thus, considerable effort and research have been invested to improve pre-training protocols and methods. The following findings resulted from this research:

- Freezing the LLM during pre-training results in acceptable zero-shot performance, but reduces in-context learning (ICL) capabilities. However, updating the LLM during pre-training encourages deep embedding alignment, which is important for effective ICL.
- Utilizing interleaved visual-language data during pre-raining is important due to its ability to update gradients accurately and retain the model's text-only capabilities.
- Integrating text-only instruction data during supervised fine-tuning (SFT) helps limit performance reduction in text-only tasks and improves accuracy in visual-language tasks.

The VILA model was developed as a unified foundation model that can integrate video, image, and language understanding and generation. As such, VILA can perform more efficiently than state-of-the-art models such as LLaVA [1]. Due to its improved pre-training process, VILA can perform multi-image reasoning, meaning that even if exposed only to single image-text pairs during SFT, VILA can perform reasoning across multiple images. The improved pre-training also exhibits enhanced ICL and world knowledge. To better understand the VILA model, we must first analyze the VLM architecture.

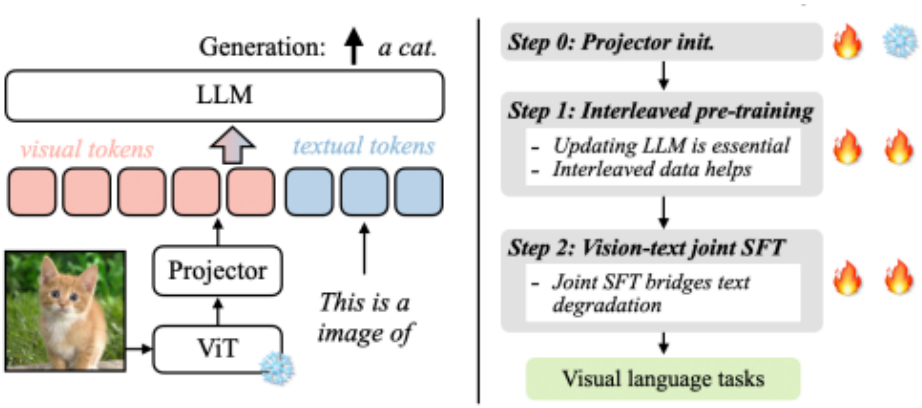

Fig. 1. Architecture of auto-regressive-based visual language models, containing the visual encoder, the projector which combines visual and textual embeddings, as well as the LLM model. The figure is copied from [2]

2) Auto-Regressive-Based VLM Architecture:

We demonstrate an example of the architecture of auto-regressive-based VLMs in Figure 1, which generate visual tokens from images and combine them with textual tokens.

As such, these combined tokens are then used as input to the model. This innovative tokenization approach effectively transforms visual information into a language-like representation, enabling the model to process and understand multimodal inputs through a unified computational framework. The architecture illustrated in Figure 1 highlights the main components of the VLM architecture: the ViT visual encoder, the projector, and the LLM, which outputs text from input images and texts. In this way, the multimodal model integrates visual and text embeddings through the projector component to align the inputs.

Figure 1 also contains information regarding the different training stages: Projector Initialization, Visual Language Pre-training, and Visual Instruction-Tuning. The first stage, Projector Initialization, refers to initializing the projector and pre-training it with random weights while the vision encoder and the LLM are frozen, thereby conducting the pre-training on image-caption pairs, focusing on aligning visual and textual embeddings as per common practice. At the Visual Language Pre-training stage, the model, including both the LLM and the projector, is trained on datasets containing both image-text pairs like COYO [3] and interleaved image-text data such as MMC4 [4]. Both these datasets were used to evaluate pre-training methodologies. This phase is computationally demanding but essential for linking the two modalities. After pre-training, the model is fine-tuned (Visual Instruction-Tuning) using datasets designed for visual language tasks. These datasets are reformatted into prompts, a style similar to FLAN [5], to adapt the model for specific tasks that involve both visual and textual information. This final fine-tuning step ensures that the model can generate coherent and accurate outputs for downstream tasks that require a deep understanding of both visual and textual inputs.

While the previous section discussed the existing protocol for pre-training focused on building general multimodal alignment and capability, as well as training aimed at refining the model for specific vision language tasks for auto-regressive-based VLMs, the VILA study explored several different pre-training methodologies. In this next section, key findings from these methodologies are presented, leading to the conclusion that fine-tuning LLMs during pre-training and instruction-tuning, as well as using a simple linear projection layer is the best solution.

3) Technical Components from Pre-Training VLMs:

The initial findings concerning the pre-training pipeline for VLMs concerns **fine-tuning vs prompt tuning**. From this study, it was observed that fine-tuning the LLM with visual inputs improved performance by enabling deeper embedding alignment and better in context learning. However, while prompt tuning, meaning the LLM is frozen and the projector is trained, avoids degrading text-only capabilities, it leads to worse generalization, especially for few-shot tasks.

With regards to **observations from running experiments**, the ablation study concluded that training only the projector results in poor performance, even with a high-capacity Transformer design. Additionally, it was found that freezing the LLM maintains 0-shot accuracy, but reduces 4-shot accuracy (ICL), particularly on out-of-distribution datasets like COCO [6] and Flickr [7]. Lastly, using a simpler linear projector instead of a Transformer forces the LLM to learn better representations, thereby resulting in improved generalization.

By calculating the Chamfer distance between visual and textual embeddings [8], and calculating the pairwise cosine similarity, the study's hypothesis that **aligning visual and textual embeddings** in deeper layers is crucial for inheriting the ICL ability of LLMs is confirmed. As such, fine-tuning the LLM achieves better alignment, demonstrated by higher cosine similarity in deeper layers and improved 4-shot accuracy.

One of the main advantages of the VILA model is its ability to function with visual data alone if needed, rather than always relying solely on combined textual and visual inputs. To support this capability, the implementation of VILA incorporates the use of an interleaved visual language corpus as a foundational dataset. As such, to properly define the visual language corpus, various datasets were evaluated to find the most suitable options for the pre-training tasks. **Interleaved datasets**, such as MMC4, are shown to contain a similar distribution to that of text-only corpora, making them effective for improving visual-language accuracy while minimizing text-only performance reduction. On the other hand, datasets based on image-text pairs, like COYO tend to degrade text-only capabilities, due to their shorter captions and reduced data diversity, which can negatively impact the model's ICL ability.

Based on experiments conducted in [2], pre-training on MMC4 was found to lead to superior visual-language accuracy and reduced degradation of text-only performance compared to COYO. Blending interleaved datasets like MMC4 with image-text pairs such as COYO enhances data diversity and downstream accuracy by leveraging the strengths of both formats. Additionally, the interleaved structure of MMC4 is especially important, as it was found that its removal resulted in significant performance decline. This highlights the importance of this structure for capturing meaningful interactions between images and text, which is essential for robust visual language learning.

The study mentions that although interleaved data improves VLM performance, it results in a small degradation of around 5% of text-only accuracy. However, this degradation can be addressed through appropriate fine-tuning. Particularly, joint supervised fine-tuning, which blends text-only instruction data with visual language data, has shown promise in restoring text-only accuracy to the level of the original LLM while also enhancing visual language task performance. This approach improves both zero-shot and few-shot capabilities, and maintains text-only performance. Additionally, blending diverse datasets, such as COYO and interleaved datasets, offers further benefits by mitigating text-only degradation and leveraging the strengths of both formats.

B. Key Elements from VILA

The final VILA model was pre-trained on 50M images and evaluated on 12 visual language benchmarks.

The **quantitative evaluation** of VILA demonstrates its strong performance on 12 visual language benchmarks, consistently outperforming state-of-the-art models. Specifically, VILA's 7B version surpasses larger competitors like LLaVA-1.5 (13B) on datasets like VisWiz [9] and TextVQA [10], thereby demonstrating the effectiveness of its pre-training process. Additionally, VILA provides advanced multi-lingual capabilities by performing better than its competitors on the MMBench-Chinese benchmark [11], while training primarily on English data. Importantly, VILA is still able to maintain its accuracy on text-only benchmarks, such as MMLU [12], BBH [13], and DROP [14], achieving comparable or better results than Llama-2 fine-tuned with text-only SFT. These results highlight VILA's ability to integrate visual language tasks while maintaining its text-only capabilities with its 13B version achieving quite high accuracy across the benchmarks.

The **qualitative evaluation** showcases VILA's ability to analyze multiple images and perform tasks that require complex visual understanding. For example, VILA can identify shared objects or themes across images, as well as distinguish differences between them, thus outperforming models like LLaVA-1.5. Additionally, VILA performs greatly at ICL, allowing it to handle tasks with few-shot examples more effectively. Another important feature is its capability for visual-chain-of-thought reasoning, where it generates detailed step-by-step explanations to solve tasks based on visual inputs, such as calculating costs or describing object relationships. VILA's pre-training also improves its world knowledge, thus allowing it to recognize famous landmarks and perform accurately in tasks involving real-world knowledge, thereby displaying its advantageous properties and usefulness in visual language applications.

II. Detailing An Efficiency Bottleneck of the VILA model

1) Introducing the Efficiency Bottleneck:

Auto-regressive VLMs, such as VILA, utilize the self-attention mechanism in the Vision Transformer (ViT) [15]. Specifically, they use Multi-Modal Self-Attention which applies to both visual and textual tokens [16]. However, the computational complexity of self-attention mechanisms scales quadratically with the input length, which can pose challenges for efficiency [17]. This complexity can be further demonstrated through the following equations of the self-attention mechanism in Transformer blocks:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

where:

- Q (queries), K (keys), and V (values) are input matrices.
- d_k is the dimensionality of the keys.

The computational complexity of self-attention can be analyzed as follows using equations from [17]:

1.1) The scaled dot-product S is computed:

$$S = \frac{QK^T}{\sqrt{d_k}} \tag{2}$$

Given that the operation involves multiplying $Q(n \times d_k)$ with $K^T(d_k \times n)$ resulting in an $n \times n$ matrix S. Therefore, the complexity is $O(n^2 \cdot d_v)$

1.2) Next, the softmax function: The softmax function is applied to each row of the matrix S, which has $n \times n$ elements or n^2. Thus, the time complexity of this process is $O(n^2)$.

1.3) The last step of the operation is to multiply the result of softmax by V: The resulting $n \times n$ matrix from the softmax operation is multiplied by $V(n \times d_v)$, yielding an $n \times d_v$ output. The time complexity from multiplying the softmax result with V is $O(n^2 \cdot d_v)$.

By summing up all these components, the total computational complexity results in:

$$O(n^2 \cdot d_k) + O(n^2) + O(n^2 \cdot d_v) = O(n^2 \cdot (d_k + d_v)) \tag{3}$$

Given that d_k and d_v are typically much smaller than n, the overall complexity simplifies to $O(n^2)$. This quadratic complexity indicates that the time required for self-attention grows proportionally to the square of the input sequence length. Therefore, to counter the efficiency bottleneck in VLMs like VILA, we propose an input-dependent reduction strategy that dynamically adjusts the model's complexity based on the input's characteristics. This solution involves a key component: Dynamic Depth Visual Encoding (DD ViT).

2) Dynamic Depth Visual Encoding (DD ViT):

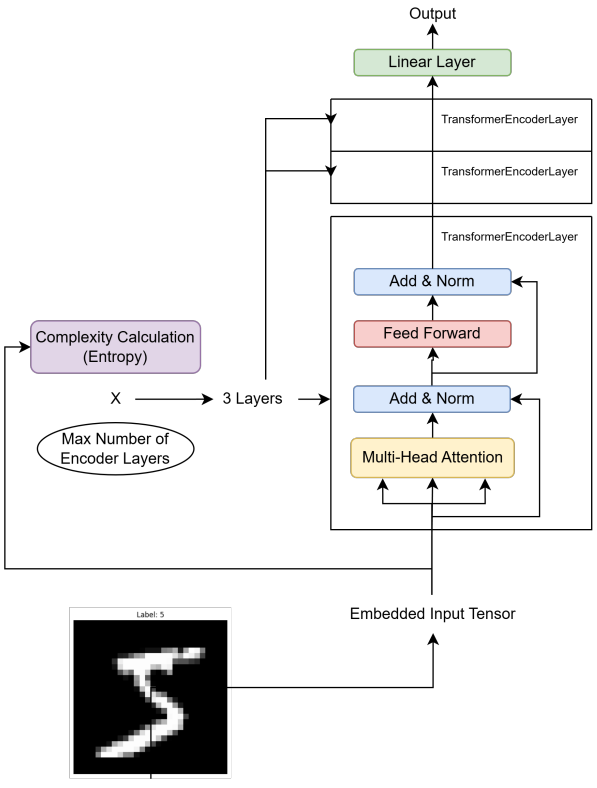

Fig. 2. ViT encoder methodology for DD ViT with complexity calculation (entropy) variance image. The handwritten digit "5" was taken from the MNIST dataset [18]. The image was made using Draw.io and the TransformerEncoderLayer was adapted from Pytorch.

This added improvement will evaluate the complexity of visual inputs and modulate the depth of processing layers. For simpler images, fewer layers are engaged, thereby reducing computational load. The first step in implementing this operation is to develop a function to analyze image complexity using entropy. Then, the visual encoder is modified to process a variable number of layers based on the assessed complexity from the function. The methodology following this is depicted in Figure 2

According to research by [19] and [20], entropy calculations can provide insights into image complexity. The entropy calculation implemented in Algorithm 1 is Shannon Entropy, which quantifies the amount of information or uncertainty in a dataset. When applied to images, it measures the unpredictability in pixel intensity values. An image with uniform pixel values, meaning all pixels are the same color, has low entropy, indicating low complexity. Conversely, an image with a wide variety of pixel values has high entropy, thereby indicating higher complexity [21]. For a batch of images, the entropy measurement is then normalized by dividing it by the maximal entropy value based for that image size, thus returning a normalized entropy value between 0 and 1. This normalized entropy is then multiplied by the maximal number of Transformer encoder layers to dynamically determine the number of layers based on the complexity of the batch of images. However, in the case of the DD ViT, the complexity is calculated on the embedded inputs

rather than the images themselves. As shown in Algorithm 1, the images first pass through the embedding layer, which maps them to a higher-dimensional space. By computing entropy on these embedded inputs, the complexity of features extracted by the embedding layer can be assessed [22]. This indirectly captures image complexity because the embedding layer is typically trained to preserve meaningful patterns in the data.

Algorithm 1 Pseudocode for the Dynamic Depth Visual Encoding

Input: Tensor x of shape $(\text{batch\_size}, depth, height, width)$
Output: Tensor of shape $(\text{batch\_size}, \text{num\_classes})$
Initialize the embedding layer: embedding $\leftarrow$ Linear$(width, \text{input_dim})$
Initialize the transformer encoder layers: encoder_layers $\leftarrow$ [TransformerEncoderLayer($d\_model = \text{input\_dim}, nhead = 4$)]$^{\text{max\_layers}}$
Initialize the fully connected layer: fc $\leftarrow$ Linear(input_dim, num_classes)
$x \leftarrow x$.squeeze(1) {Remove channel dimension $(\text{batch\_size}, height, width)$}
$x \leftarrow x$.permute$(0, 2, 1)$ {Reshape x to $(\text{batch\_size}, \text{len\_sequence} = width, \text{input\_dim})$}
$x \leftarrow$ embedding(x) {Embedding the layer}
$entropy \leftarrow -\text{sum}(prob\_dist \times \log_2(prob\_dist + 1e - 9))$ {Compute the chosen complexity metric, Shannon Entropy}
Normalize entropy values to range $[0, 1]$: $input\ complexity \leftarrow entropy \div maximal\ entropy$ {Divide entropy by maximal entropy}
$number\_layers \leftarrow \text{clamp}(\text{floor}(complexity \times \text{max\_layers}), 1, \text{max\_layers})$
for $i = 1$ to max_layers **do**
  if $i > \max(number\_layers)$ **then**
    break
  end if
  $x \leftarrow$ encoder_layers$[i - 1](x)$ {Apply the encoding layer}
end for
$x \leftarrow \text{mean}(x, \text{dim} = 1)$ {Perform global average pooling over the sequence dimension}
$output \leftarrow \text{fc}(x)$ {Apply the fully connected layer}

return $output$

III. Experiments and Results of Dynamic Depth Visual Encoding

A. Setting up the experiment

To demonstrate the effectiveness of this improved methodology serving as the visual encoding step, the dynamic depth visual encoding method was compared to a baseline transformer model using metrics such as test accuracy and floating-point operations per second on the MNIST dataset. As such, the MNIST dataset contains 8-bit grayscale images of handwritten digits from 0 to 9. The training set is composed of 60,000 images, and the testing of 10,000 images. Each image is 28x28 pixels, thus resulting in 784 pixels per image. The images are in grayscale, with pixel intensity values ranging from 0 to 255 [18].

The baseline visual encoding model is comprised of an embedding layer, as well as four transformer encoder layers, and a fully connected layer. Each transformer encoder layer encompasses self-attention mechanisms and feedforward neural networks as shown in 2. After encoding, the model applies global average pooling to aggregate information across the sequence, resulting in a fixed-size representation. That representation is then passed through a fully connected layer to generate the final class predictions.

The Dynamic Depth Visual Encoding (DD ViT) model is based on the baseline model, featuring an adaptive mechanism that adjusts its processing depth based on the entropy of the input images. It starts by embedding each 28-dimensional input feature into a higher-dimensional space using a linear layer. Then, the model computes the embedded input complexity by calculating the Shannon Entropy and normalizing these values against the maximal possible entropy for this type of input. In the algorithm, the embedded inputs are, for the entropy calculation, normalized from 0 to 1 and are mapped to a histogram with 256 bins. Thus, the maximal entropy value can be calculated following [23]:

$$H = -\sum_{i=0}^{255} P(i) log_2 P(i) = -\sum_{i=0}^{255} \frac{1}{256} log_2 \frac{1}{256} = 8\ bits \tag{4}$$

Then, we normalize the entropy by dividing by 8 bits. The normalized complexity determines the number of Transformer encoder layers, that the input will traverse, allowing the model to allocate more computational resources to complex inputs and less to simpler ones.

B. Results

The results from running the baseline ViT model against the Dynamic Depth Visual Encoding model are reported in I. Both models were run with the same parameters for the same amount of epochs.

TABLE I
THE FLOATING-POINT OPERATIONS PER SECOND AND TEST ACCURACY ON MNIST DATASET FROM THE BASELINE VISUAL ENCODING MODEL AND THE DYNAMIC DEPTH (DD) VISUAL ENCODING MODEL.

Model	FLOPS	Accuracy
Baseline ViT	2.004 GFLOPs	95.31
DD ViT	1.504 GFLOPS	95.24

As it can be observed, the baseline ViT model achieved a testing accuracy score of 95.31%, while the DD ViT model managed to reach 95.24%. However, it is important to notice that the DD ViT model was able to reduce the number of floating-point operations per second (FLOPs) by 0.5 GFLOPs. This efficiency gain is attributed to the number of Transformer encoder layers: the baseline ViT model uses a fixed four-layer configuration, whereas the DD ViT model adjusts the number of layers dynamically based on input complexity. For most inputs, the DD ViT required only three encoder layers, while others required only two encoder layers. However, the algorithm determines the number of layers per batch based on the maximum complexity within that batch. For example, if a batch of 64 embedded inputs includes 40 that require two Transformer encoder layers and 24 with higher complexity that require three layers, the batch is processed using three layers. By observing some of the outputted entropy scores, it is clear that the majority of the inputs require three Transformer encoder layers, thus explaining the reduced floating-point operations per second for the DD ViT. This approach can potentially provide an effective solution, as it significantly reduces computational complexity while resulting in minimal accuracy reduction.

C. Limitations

Although the Shannon Entropy can provide information on the uncertainty and randomness within an image by capturing the distribution of pixel intensities which can be extended as a measure for image complexity, it does not account for spatial relationships between pixels as mentioned by [24]. This means that two images that have the same histogram, will be considered to have the same entropy values, even if their spatial arrangements differ which would also impact their complexities. For future implementations, looking into delentropy [21] or multiscale entropy (MSE) [25] could be a potential solution to bypass this limitation.

IV. CONCLUSION

In the initial section of this report, we first conducted a thorough review of the VILA model and its extensions. This comprehensive study discussed VILA's core characteristics, identified significant technical innovations, and pinpointed potential areas for improvement. As such, VILA demonstrates strong multilingual capabilities, it can maintain competitive accuracy on text-only benchmarks. VILA can also perform accurately processing multiple images, identifying objects or themes, distinguishing differences, and excelling in ICL and visual chain-of-thought reasoning. Its pre-training also enhances work knowledge. In conclusion, VILA provides an improvement to Visual Language Models due to the optimized pre-training methodology.

In the second part of the report, we analyzed an efficiency bottleneck that impacts VLMs and the VILA model architecture, specifically the quadratic time complexity introduced by the self-attention mechanism in the Vision Transformer (ViT) architecture. To address this, we proposed the implementation of a Dynamic Depth Visual Encoding (DD ViT) model designed to reduce this complexity. We also validated the performance of this model by comparing it against a baseline ViT model using the MNIST dataset and recording both testing accuracy and floating-point operations per second. While the testing accuracy of both the baseline model and the DD ViT model is quite similar, the DD ViT model was able to reduce the complexity by dynamically adapting the number of Transformer encoder layers used in the model. These findings could inspire more improvements to VILA and lay the groundwork for creating better and more efficient Visual Language Models in the future.

DECLARATION OF NO PLAGIARISM

I hereby declare that this report is an original work. I have independently implemented all algorithms and drafted all content myself. Any sources referenced in this report are properly cited and credited.

I acknowledge that the inclusion of source code result in a higher match rate; however, this does not detract from the originality of my analysis and findings.

REFERENCES

[1] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," 2024. [Online]. Available: https://arxiv.org/abs/2310.03744

[2] J. Lin, H. Yin, W. Ping, Y. Lu, P. Molchanov, A. Tao, H. Mao, J. Kautz, M. Shoeybi, and S. Han, "Vila: On pre-training for visual language models," *Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26 679–26 689, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:266174746

[3] C.-Z. Lu, X. Jin, Q. Hou, J. H. Liew, M.-M. Cheng, and J. Feng, "Delving deeper into data scaling in masked image modeling," 2023. [Online]. Available: https://arxiv.org/abs/2305.15248

[4] W. Zhu, J. Hessel, A. Awadalla, S. Y. Gadre, J. Dodge, A. Fang, Y. Yu, L. Schmidt, W. Y. Wang, and Y. Choi, "Multimodal c4: An open, billion-scale corpus of images interleaved with text," 2023. [Online]. Available: https://arxiv.org/abs/2304.06939

[5] S. Longpre, L. Hou, T. Vu, A. Webson, H. W. Chung, Y. Tay, D. Zhou, Q. V. Le, B. Zoph, J. Wei, and A. Roberts, "The flan collection: Designing data and methods for effective instruction tuning," 2023. [Online]. Available: https://arxiv.org/abs/2301.13688

[6] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015. [Online]. Available: https://arxiv.org/abs/1405.0312

[7] J. McAuley and J. Leskovec, "Image labeling on a network: Using social-network metadata for image classification," 2012. [Online]. Available: https://arxiv.org/abs/1207.3809

[8] T. Wu, L. Pan, J. Zhang, T. Wang, Z. Liu, and D. Lin, "Density-aware chamfer distance as a comprehensive metric for point cloud completion," 2021. [Online]. Available: https://arxiv.org/abs/2111.12702

[9] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham, "Vizwiz grand challenge: Answering visual questions from blind people," 2018. [Online]. Available: https://arxiv.org/abs/1802.08218

[10] S. Hegde, S. Jahagirdar, and S. Gangisetty, "Making the v in text-vqa matter," 2023. [Online]. Available: https://arxiv.org/abs/2308.00295

[11] Y. Liu, H. Duan, Y. Zhang, B. Li, S. Zhang, W. Zhao, Y. Yuan, J. Wang, C. He, Z. Liu, K. Chen, and D. Lin, "Mmbench: Is your multi-modal model an all-around player?" 2024. [Online]. Available: https://arxiv.org/abs/2307.06281

[12] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," 2021. [Online]. Available: https://arxiv.org/abs/2009.03300

[13] M. Suzgun, N. Scales, N. Schärli, S. Gehrmann, Y. Tay, H. W. Chung, A. Chowdhery, Q. V. Le, E. H. Chi, D. Zhou, and J. Wei, "Challenging big-bench tasks and whether chain-of-thought can solve them," 2022. [Online]. Available: https://arxiv.org/abs/2210.09261

[14] D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, and M. Gardner, "Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs," 2019. [Online]. Available: https://arxiv.org/abs/1903.00161

[15] H. V. Koay, J. H. Chuah, C.-O. Chow, and Y.-L. Chang, "Detecting and recognizing driver distraction through various data modality using machine learning: A review, recent advances, simplified framework and open challenges (2014–2021)," *Engineering Applications of Artificial Intelligence*, vol. 115, p. 105309, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197622003517

[16] Y. Li, J. Wang, X. Dai, L. Wang, C.-C. M. Yeh, Y. Zheng, W. Zhang, and K.-L. Ma, "How does attention work in vision transformers? a visual analytics attempt," 2023. [Online]. Available: https://arxiv.org/abs/2303.13731

[17] F. Duman Keles, P. Mahesakya Wijewardena, and C. Hegde, "On the computational complexity of self-attention," 2022. [Online]. Available: https://arxiv.org/abs/2209.04881

[18] Y. B. Y. LeCun, L. Bottou and P. Haffner, "Gradient-based learning applied to document recognition," 1998.

[19] H. Zhang, J. Dong, S. He, and S. Lv, "Research on image complexity description method based on approximate entropy," pp. 918–920, 2020.

[20] A. A. Rahane and A. Subramanian, "Measures of complexity for large scale image datasets," p. 282–287, Feb. 2020. [Online]. Available: http://dx.doi.org/10.1109/ICAIIC48513.2020.9065274

[21] K. G. Larkin, "Reflections on shannon information: In search of a natural information-entropy for images," 2016. [Online]. Available: https://arxiv.org/abs/1609.01117

[22] A. Perevalov, D. Kurushin, R. Faizrakhmanov, and F. Khabibrakhmanova, "Question embeddings based on shannon entropy - solving intent classification task in goal-oriented dialogue system," 2019. [Online]. Available: https://opendata.uni-halle.de//handle/1981185920/13572

[23] S. Vajapeyam, "Understanding shannon's entropy metric for information," 2014. [Online]. Available: https://arxiv.org/abs/1405.2061

[24] H. Yu and S. Winkler, "Image complexity and spatial information," *2013 Fifth International Workshop on Quality of Multimedia Experience (QoMEX)*, pp. 12–17, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:2677022

[25] A. Humeau-Heurtier, "Multiscale entropy approaches and their applications," *Entropy*, vol. 22, no. 6, 2020. [Online]. Available: https://www.mdpi.com/1099-4300/22/6/644