Deformable Attention Perceivers

Angel Ontiveros
angel.ontiveros@studium.uni-hamburg.de

Computer Vision Master Project
Department of Informatics, University of Hamburg, Germany
11.09.2023

***Abstract*—Multimodal learning is one of the current hot topics in Artificial Intelligence research. It better approximates how humans process their environment by making use of multiple senses. Moreover, it has the benefits of increased robustness and possibly increased task performance. However, this learning regime faces multiple challenges, such as finding a suitable multimodal fusion architecture that can achieve acceptable performance for the task at hand, finding universal architectures that can work on a vast range of tasks, and addressing the increased computational costs that may be present in multimodal learning due to increased data volumes, and number of parameters needed. Some recent architectures such as Perceivers are able to process any data modality or combination of modalities efficiently, while some techniques such as deformable attention have shown increased task performance without significantly increasing computational costs. In this paper, we explore combining these two ideas to yield a system that can process any modality or combination of modalities efficiently with the Perceiver while also increasing its task performance using deformable attention. We observe that it is possible to increase task performance with this approach but additional experiments on more datasets and modalities are still required to make a more certain statement.**

I. Introduction

Humans and many living creatures perceive their environment through multiple senses, such as vision, audition, touch, smell, and taste. These input modalities are used in day-to-day life to make decisions. Often not each on their own, but combined. A typical example is when having a face-to-face conversation, we not only listen to the other person, but see the movement of their lips, face, and body. We can integrate this information to better understand the meaning conveyed. In contrast, deep learning systems for many years were exclusively focused on learning from single modalities, such as text or images. Now the field is moving to learn from multiple modalities or senses simultaneously to reap the benefits we can observe in biological systems that do this, such as increased task performance and robustness. Furthermore, one can think of multiple applications that require the processing of multiple modalities, such as generative models that work at the same time with text and images, different types of robots, autonomous vehicles, and medical data analysis, just to name a few.

A. Challenges

In the current literature on multimodal learning, several challenges and limitations are mentioned, some works such as [1], [17], and [7] have surveyed the field and compiled lists of challenges. In this work, we are interested in addressing the following challenges:

- Fusion strategy: How to fuse modalities and achieve similar or even improved results in terms of task performance?
- Computational costs: When working with multiple modalities, the raw volume of data may increase, by having for example images and audio, instead of just images for a given task. Learning algorithm space and time complexity depend on the number of data points to be processed and their dimensionality, making multimodal learning require increased amounts of computation and memory. Therefore, implementing computationally efficient multimodal learning architectures is paramount.
- Universalness: Typically in multimodal systems, network portions that can effectively handle distinct modalities and/or tasks have to be designed separately. The aforementioned is done because different inductive biases are useful in different types of data. For image data, Convolutional Neural Network (CNN) architectures are mostly used, while in sequential data such as text or audio Recurrent Neural Network (RNN) architectures have dominated in the last few years. Furthermore, these modalities have to be fused, which leads to complex and specialized multimodal learning architectures. For this reason, a simpler, more universal approach is desirable.

B. Objectives

The main objective of this research is to answer what is the impact on task performance of adding deformable attention [19] in multimodal systems that exhibit the universalness characteristic. To achieve this, we need to assess current multimodal systems that exhibit universalness characteristics.

For this, we select the Perceiver architecture ([6], [5]) as the baseline. We also need to properly understand and integrate deformable attention techniques into our base approach. Moreover, suitable datasets have to be selected, gathered, and processed for testing our resulting system. Finally, experiments need to be carried out to assess the effect on task performance of adding deformable attention to Perceivers.

C. Contributions

The main contribution of this research is integrating the deformable attention mechanism described by [16] in Perceiver architectures ([6], [5]), as well as evaluating the effect of such an integration. Furthermore, we assess at what depth level of the Perceiver's architecture multiple self-attention and cross-attention blocks it might be more beneficial to incorporate deformable attention. According to [16], it might make more sense to incorporate deformable attention not so much at the beginning, but in the last stages of the processing pipeline, since deformable attention looks at what is important in the big picture and typically early stages capture more low-level local patterns. By comparing to regular Perceiver baselines, we are able to see the effects of deformable attention in this kind of multimodal system architecture and gain insights into whether this technique can improve task performance on systems that perform early fusion and exhibit universalness characteristics without major increases in computational costs.

D. Evaluation

For evaluation of the results against the selected baselines, such as Perceiver models and deformable attention as described in [16], a similar dataset to those used in the original papers for image classification is used. Similarly, we use Top-1 Accuracy metric as is used in the baselines for comparison, plus some additional ones, such as Precision, Recall, and F1-score.

II. Related work

A. Multimodal Learning

To pose the multimodal learning problem, let's follow the explanation given by [1]. For simplicity, consider two modalities, namely audio and video. For these modalities, we have input streams $\mathbf{X_a} = \{x_1^n, \ldots, x_T^n\}$ and $\mathbf{X_v} = \{x_1^m, \ldots, x_T^m\}$ where n and m represent the dimension of the vectors for each modality and $1 \ldots T$ are the time steps at which these vectors occur. As an example, x_1^n and x_1^m occur at the same time step but may have different dimensions. Now suppose we have one neural network for each of these modalities $\mathbf{N_a} : \mathbf{X_a} \rightarrow Z$ and $\mathbf{N_v} : \mathbf{X_v} \rightarrow Z$ that maps these modalities to a ground-truth label $Z = \{Z^1 \ldots Z^T\}$. Then, we would like to have a network $\mathbf{M} = \mathbf{N_a} \oplus \mathbf{N_v}$ that can map to the ground truth labels by fusing or combining both modalities, here the $\oplus$ operator stands for fusion. This leads us to the problem of how to fuse the modalities. As explained by [1] **Fusion algorithms** can be broadly categorized in three approaches:

- *Early Fusion*: refers to combining information at the data level, for example, here we would combine X_a and X_v data through means of concatenation, summation, linear combinations, or other functions, and then process them together.
- *Late Fusion*: here we would process modalities separately with networks N_a and N_v and have the outputs combined in the end, by a sum, weighted sum, or other functions.
- *Hybrid Fusion*: This is an in-between of the previous two and can consist of combining features at an intermediate level, or at multiple levels. For example, combining raw input, some intermediate representations in the network, and the output of the networks.

Several techniques and functions have been explored for combining modalities, [1] mentions kernel-based approaches, probabilistic graphical model approaches, and deep learning methods such as deep belief networks, autoencoders, convolutional and recurrent neural networks, as well as attention mechanisms. Some of the best-performing techniques in actuality are based on neural networks and attention mechanisms with transformers [14], [4].

B. Perceiver

Previous works that will be used as theoretical foundations for this research are on one hand the Perceiver works ([6], [5]), since their approach strives to be as general as possible in the choice of modality or combination of modalities it is able to process. This makes the Perceiver exhibit the universalness characteristic as defined by [17].

The authors published two papers, Perceiver and PerceiverIO, with PerceiverIO being an extension of the original Perceiver architecture. The main idea of this architecture is to be able to handle and learn from any modality or combination of modalities of data without needing to make major changes to the architecture to accommodate the modalities. To achieve this, the authors leverage the attention mechanisms provided by transformer models, which correlate and give weights to each token in the inputs with respect to every other token, a characteristic that makes transformers potentially applicable to any type of input. To handle any input modality, a preprocessing step consisting of a mapping to a simple 2D array is done, where the rows of this 2D array can be representations of pixels, image patches, words, characters, or even a concatenation of multiple modalities. The main issue in applying transformers as described is that they can be computationally expensive. Each stage's time complexity scales quadratically with the input data size. For example, if the input data consists of M vectors the time complexity is $\mathcal{O}(M^2)$, and taking into account that an architecture consists of L layers, then the total complexity is $\mathcal{O}(LM^2)$. As explained by [6] the main attention operation that results in the aforementioned complexity is:

$$softmax\left(QK^T\right)V$$

Where $Q \in \mathbb{R}^{M\times D}$, $K \in \mathbb{R}^{M\times C}$, and $V \in \mathbb{R}^{M\times C}$ (C and D are each vector dimensions). To reduce the attention operation complexity, [6] replaced Q by a projection from a latent learned array, changing the dimensions to $Q \in \mathbb{R}^{N\times D}$, where $N \ll M$. Considering L layers, since we have to multiply with the input data dimensions only on the input layer, this results in a total complexity of $\mathcal{O}(MN + LN^2)$.

Furthermore, the PerceiverIO architecture [5] extends the Perceiver to also be able to have flexible outputs, so not only handle any input modality but give any output size required. This is achieved by adding a transformer decoder component to the Perceiver, using cross-attention the decoder creates K and V matrices from the output of the encoder, while the Q matrix is designed to shape the desired output. PerceiverIO architecture can be seen in Figure 1.

This architecture effectively ensures that the transformer blocks processing depends only on the latent array size and not on the input data size, which in turn reduces computational costs and frees the model from dealing with input modalities' specific dimensions.

Perceivers' task performance results are close to those of other modality-specific architectures, nevertheless, sometimes below state-of-the-art (SOTA).

C. Deformable Attention

This research makes use of literature discussing modern deformable attention methods in transformers, in particular, the work done by [16]. Here, the authors work with visual data (images). The main weaknesses of previous approaches that this paper addresses are 1) that using a global receptive field such as in [3] results in high memory and computational costs, and 2) that using sparse attention such as in [15], and [8] may result in failing to capture long-range dependencies. To alleviate these issues the proposed deformable attention module can be used as a transformer building block (i.e. a layer) to select the position of keys and values in a data-dependent way, which are later used to compute the transformer self-attention.

In contrast to other approaches that calculate each element's offset separately the selected offsets/positions are shared for all queries. This was done based on the observations done by [2], and [18] that in global attention different queries tend to yield almost the same attention pattern.

Consider an image as input with dimensions $x \in \mathbb{R}^{H\times W\times C}$ The image is sub-sampled by generating a grid dividing the height and width of the image by a factor r, this generates a set of reference points $p \in \mathbb{R}^{H_G\times W_G\times 2}$ on the grid, such that $H_G = H/r, W_G = W/r$. These points are linearly spaced coordinates $\{(0,0),\ldots,(H_G-1,W_G-1)\}$ which later get mapped to a range $[-1,+1]$. Using a sub-network that is integrated into regular transformer blocks they learn to generate offsets for the points on the grid from the queries $\Delta p = \theta_{\text{offset}}(q)$, which then are used to deform these points and select which parts of the image should be paid attention to, an illustration of this can be seen in Figure 2. While the details of the offset-generating sub-network are illustrated in Figure 3 **(b)**. To stabilize the range of the offsets during training a constant s is used, such that $\Delta p \leftarrow s\tanh(\Delta p)$.

Then features are sampled at the locations of the deformed points according to the following equations:

$$q = xW_q, \tilde{k} = \tilde{x}W_k\tilde{v} = \tilde{x}W_v$$
$$\Delta p = \theta_{\text{offset}}(q)$$
$$\tilde{x} = \phi(x; p + \Delta p)$$

Where $\tilde{x}$ are the deformed extracted features, and $\tilde{k}, \tilde{v}$ are the deformed keys and values respectively.

$\phi(\cdot;\cdot)$ is a bilinear interpolation sampling function, defined as follows:

$$\phi\left(z;(p_x,p_y)\right) = \sum_{(r_x,r_y)} g\left(p_x,r_x\right) g\left(p_y,r_y\right) z\left[r_y,r_x,:\right]$$
$$g(a,b) = \max(0,1-|a-b|)$$

With (r_x, r_y) indexing the locations of the input $z \in \mathbb{R}^{H\times W\times C}$.

Next, multi-head attention scores are calculated, with the distinction of adding a position bias $\phi(\hat{B};R) \in \mathbb{R}^{HW\times H_GW_G}$.

$$z^{(m)} = \sigma\left(q^{(m)}\tilde{k}^{(m)\top}/\sqrt{d} + \phi(\hat{B};R)\right)\tilde{v}^{(m)},$$

$\sigma(\cdot)$ is the softmax function. All attention heads' outputs are concatenated and projected by output weights W_o.

$$z = \text{ Concat }\left(z^{(1)},\ldots,z^{(M)}\right)W_o,$$

Finally, as is usual in transformer architectures Layer Normalization (LN) is used and the output from the previous block $l-1$ is summed with the result of the current multi-head attention and then this intermediate result is passed through a Multilayer Perceptron (MLP):

$$z'_l = \text{MHSA}\left(\text{LN}\left(z_{l-1}\right)\right) + z_{l-1}$$
$$z_l = \text{MLP}\left(\text{LN}\left(z'_l\right)\right) + z'_l$$

The complete inner workings of a Deformable Attention Transformer (DAT) module are shown in Figure 3.

III. Methods

As explained in the related work section, Perceivers can in theory process any modality or combination of modalities, given they are reshaped into a 2D format. This is the desirable universalness characteristic. Nonetheless, their task performance lags behind modality-specific architectures.

One probable way of increasing Perceivers' task performance could be by using deformable attention methods,

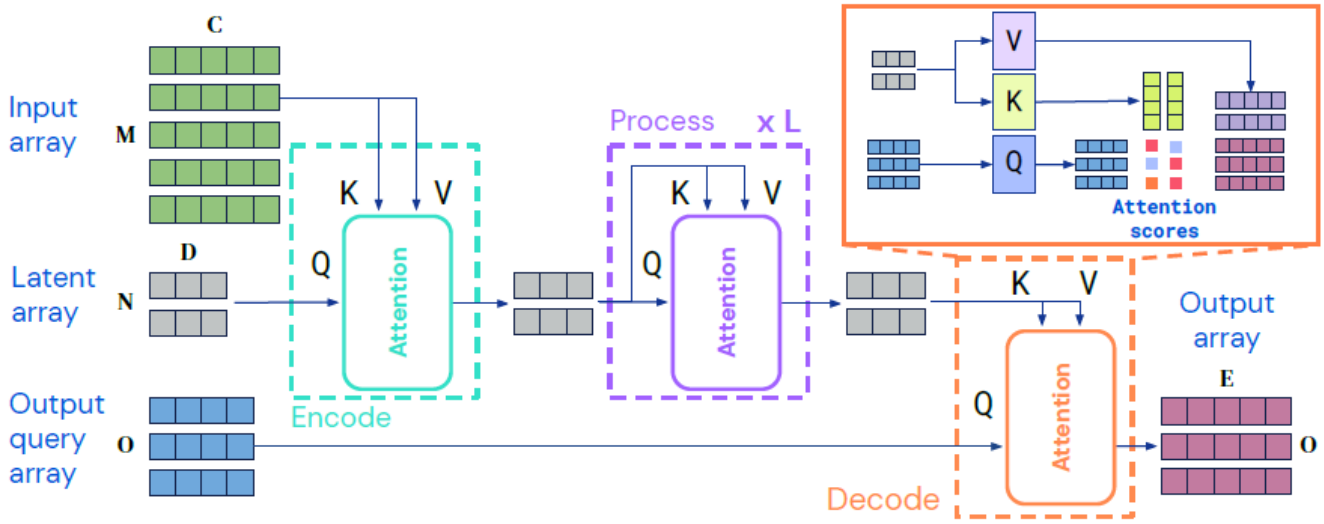


Figure 1: In the PerceiverIO architecture, a latent trainable array of defined dimensions is initialized, from this latent array query values are extracted and used to calculate attention scores, which effectively makes further processing by attention blocks depend only on the size of the latent array and not the input data. In the decoder, the output shape can be defined similarly by generating an output query array. adapted from [5].

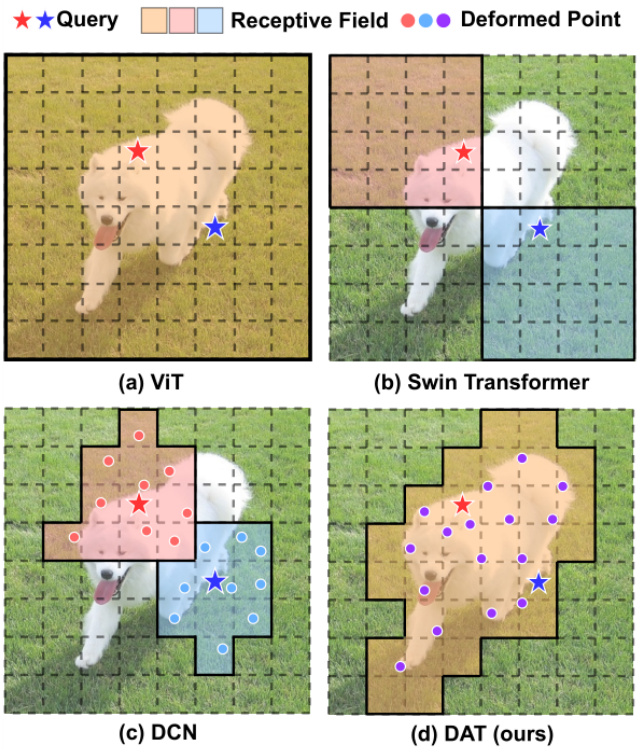


Figure 2: DAT attention pattern compared to other methods, adapted from [16].

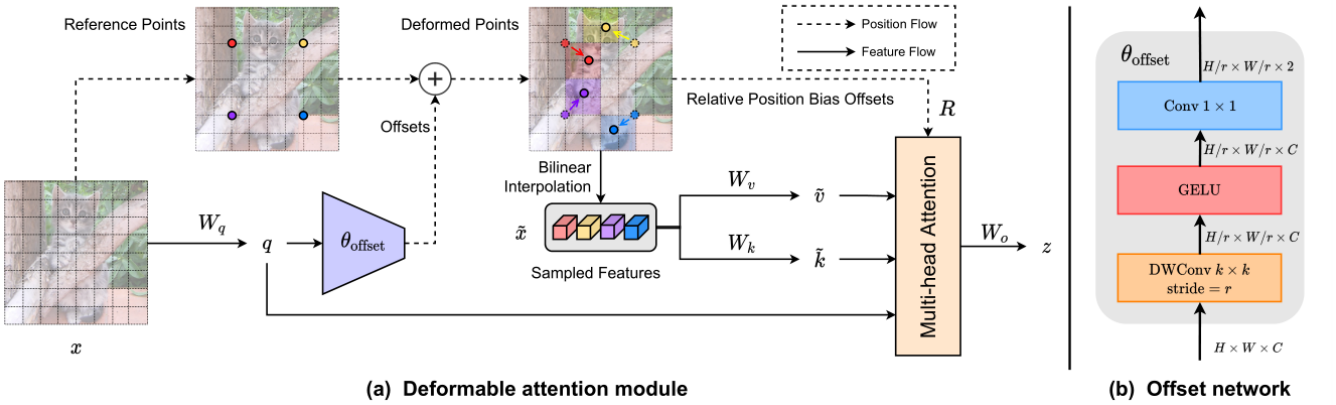
as they have previously been successful in increasing task performance in visual domain tasks, as shown in [16]. To the best of our knowledge, there is no previous research investigating the impact of deformable attention in Perceiver architectures, and in particular, using the deformable attention flavor proposed in [16].

Examining closely both approaches it becomes apparent that it is technically feasible to combine them. This is because Perceivers in their processing use regular attention blocks and map any modality or combination of modalities to a latent representation consisting of a 2D feature map. At the same time, the deformable attention method as proposed by [16], was evaluated on 2D color images and consists of a transformer block module with an embedded sub-network that trained jointly to calculate the offsets to then get deformed points and then corresponding deformed keys and values to calculate attention scores.

Therefore, in this paper, we integrate the deformable attention module in the Perceiver architecture to attempt to increase task performance. Since both the proposed deformable attention and the Perceiver architecture have been evaluated in vision tasks, in particular image classification, using the ImageNet-1K dataset, in this work we use a similar dataset.

A. Dataset

To implement a similar approach in image classification images as those done in [5] and [16] We use the Tiny-ImageNet dataset available at Hugging Face platform [12]. The dataset consists of 100.000 color images for training, 200 for each class, and an additional 10.000 for testing, 50 for each class. The training dataset was split into 90.000 images



(a) Deformable attention module **(b) Offset network**

Figure 3: DAT module internal mechanism, (a) is a deformable attention module and (b) is the offsets generation network. adapted from [16].

for training and 10.000 images for validation, with a split factor of 0.1.

During the implementation, it was noted that some of the images were not color images and therefore they had to be filtered out from the dataset to appropriately work with the implemented Perceiver architecture. After filtering, the dataset consisted of 88366 training images, 9813 validation images, and 9832 testing images.

B. Architecture

We use one of the implementations of the Perceiver for computer vision tasks available at Hugging Face [13]. Furthermore, following the findings of [16] regarding at what stage it is more beneficial to use Deformable attention, we introduce the deformable attention module at the end of the transformer pipeline. We use the same parameters configuration for the deformable attention as used in the original paper's base version, which includes a patch size of 4, 32 attention heads, and an embedding/latent representation of size 1024. For the implementation in Python, PyTorch framework with the Pytorch Lightning library was used. The associated code can be found on the following GitHub repository [11].

C. Training

All Models were trained on Google Colab [10] with 51GB System RAM, and A100 GPUs with 16GB VRAM for 6 Epochs using the Tiny-ImageNet dataset, 16-bit precision training, AdamW optimizer [9], a learning rate of $5e^{-4}$, and a batch size of 20. The batch size and learning rate were found using Pytorch Lightning's Tuner library.

The training was started from the existing pre-trained weights in the Hugging Face Perceiver model [13]. In order to reduce the training time and computational costs required for each experiment most of the Perceiver's encoder and decoder layers were frozen and only the last layer and newly added layers were trained.

IV. Experiments

Evaluations are done against Perceiver [6], [5] baselines using similar datasets and metrics for image classification as in the original paper.

Three models were trained, (1) a vanilla baseline Perceiver in which only the last layer (Classification head) is replaced to adapt the existing model to the Tiny-ImageNet dataset which contains 200 distinct classes. (2) A Perceiver model is augmented with an intermediate Fully Connected (FC) layer with 1024 nodes and then the classification head is adapted for 200 classes, as in the previous model. (3) A deformable attention block as specified in [16] base version is added in between the FC layer and the adapted classification head.

We then proceed to evaluate and compare these three models using Top-1 Accuracy, Precision, Recall, and F1-score metrics in order to assess whether there is a significant difference in task performance between a vanilla Perceiver, a Perceiver augmented with a single new fully connected layer and a Perceiver augmented with deformable attention.

For reference, Perceiver results on the ImageNet dataset are shown in Figure 4. Similarly results for the original Deformable Attenion Transformer (DAT) are shown in Figure 5.

ResNet-50 (He et al., 2016)	77.6
ViT-B-16 (Dosovitskiy et al., 2021)	77.9
ResNet-50 (FF)	73.5
ViT-B-16 (FF)	76.7
Transformer (64x64, FF)	57.0
Perceiver (FF)	78.0

Figure 4: Perceiver Accuracy comparison on ImageNet dataset, adapted from [6].

ImageNet-1K Classification				
Method	Resolution	FLOPs	#Param	Top-1 Acc.
DeiT-S [33]	224^2	4.6G	22M	79.8
PVT-S [36]	224^2	3.8G	25M	79.8
GLiT-S [5]	224^2	4.4G	25M	80.5
DPT-S [7]	224^2	4.0G	26M	81.0
Swin-T [26]	224^2	4.5G	29M	81.3
DAT-T	224^2	4.6G	29M	**82.0** (+0.7)
PVT-M [36]	224^2	6.9G	46M	81.2
PVT-L [36]	224^2	9.8G	61M	81.7
DPT-M [7]	224^2	6.9G	46M	81.9
Swin-S [26]	224^2	8.8G	50M	83.0
DAT-S	224^2	9.0G	50M	**83.7** (+0.7)
DeiT-B [33]	224^2	17.5G	86M	81.8
GLiT-B [5]	224^2	17.0G	96M	82.3
Swin-B [26]	224^2	15.5G	88M	83.5
DAT-B	224^2	15.8G	88M	**84.0** (+0.5)
DeiT-B [33]	384^2	55.4G	86M	83.1
Swin-B [26]	384^2	47.2G	88M	84.5
DAT-B	384^2	49.8G	88M	**84.8** (+0.3)

Figure 5: Deformable Attention transformer accuracy comparison with other models on ImageNet dataset, adapted from [16].

V. Results

In Table I are the results for the metrics in the validation set for all three trained models, the baseline Perceiver with a new classification head attached, an Ablation Perceiver with one additional FC layer of 1024 added before the classification head, and finally the Deformable Attention Perceiver. We observe that for all metrics in the validation set adding only a fully connected layer decreased significantly performance across all metrics. Nonetheless, the Deformable attention Perceiver performed better than the baseline model in an amount similar to performance gains reported by [16] in the original DAT paper.

In addition, in Table II we can see the corresponding results for the test set. Here the ablation model with an added FC layer also performed significantly worse than the baseline model. While the Deformable Attention Perceiver model performed slightly worse than the baseline.

It is worth noting that metric values are a few percent below those reported in the original Perceiver [13], this is due to training on only a few epochs.

From these results on the validation set, we can infer that it is possible to slightly increase performance using deformable attention, although it is unclear why results on the test set performed slightly worse than the baseline, additional experiments with other datasets and perhaps other modalities would be necessary to explore further.

Validation			
Metric	**Baseline**	**Ablation FC**	**Deformable**
Accuracy	70.586%	67,9894% **(-2,5966)**	71,1076% **(+0,5216)**
Precision	72,343%	70,1329% **(-2,2101)**	72,9863% **(+0,6432)**
Recall	70,586%	67,9894% **(-2,5966)**	71,1076% **(+0,5216)**
F1-score	70,1918%	67,4065% **(-2,7853)**	70,8199% **(+0,6281)**

Table I: Validation metrics for Baseline Perceiver, Ablation adding a fully connected layer (FC), and Deformable Attention Perceiver. Numbers in parenthesis and bold are the percentual difference with respect to the baseline.

Test			
Metric	**Baseline**	**Ablation FC**	**Deformable**
Accuracy	69,7785%	67,629% **(-2,1494)**	69,4314% **(-0,3470)**
Precision	72,0995%	70,1869% **(-1,9126)**	71,9367% **(-0,1628)**
Recall	69,7785%	67,629% **(-2,1494)**	69,4314% **(-0,3470)**
F1-score	69,6829%	67,3791% **(-2,3038)**	69,3125% **(-0,3704)**

Table II: Test metrics for Baseline Perceiver, Ablation adding a fully connected layer (FC), and Deformable Attention Perceiver. Numbers in parenthesis and bold are the percentual difference with respect to the baseline.

VI. Limitations

A. Training Resources

Given the computational resources and time available, training of each of the models was done for 6 epochs, in contrast to training for 100+ epochs that were done in [6], [5] and, [16]. This made our model not reach the same level of accuracy as in the mentioned works. Nonetheless, reasonable performance was achieved and a comparison could be done between the models.

B. Multimodality

In this paper, we analyzed the performance of the Deformable Attention Perceiver using only the visual modality, i.e. color images. As has been shown in the Perceiver and PerceiverIO papers, this architecture can in theory be applied to any modality or combination of modalities. Furthermore,

As was explained in this paper the deformable attention module proposed by [16], could in theory be applied to any modality under this architecture too. Future research could include evaluating Derformable Attention Perceivers in multiple distinct modalities (audio, text, point clouds) or combinations of modalities such as in videos, captioned images, robot multisensor data, medical data, etc.

C. Preprocessing and Postprocessing

Although the proposed architecture can in theory process and perform tasks on any modality, it is still necessary to have a preprocessing layer, and depending on the task, a postprocessing layer to shape the data in the expected format for inputs and outputs. In the PerceiverIO architecture [5], the requirement for the inputs is that whatever input modalities are used, they need to be formatted into a 2D matrix so that queries can be used with the latent's array keys and values, this may imply increasing or decreasing the number of dimensions in the original data format. The output format and shape can be specified through the transformer decoder query. Additional postprocessing may be required depending on the task.

D. Evaluated tasks

The Perceiver architecture can be considered to have the universalness characteristic as described by [17], which means it could handle any modalities and tasks. However, in this paper, we only analyzed task performance on an image classification task. One possible future research could be evaluating Deformable Attention Perceivers architecture on other tasks, such as generative tasks in different modalities.

E. Interpretability

One of the challenges in multimodal learning and transformers in general as explained by [17] is interpretability. This is particularly important in areas where stakes are high and human evaluation and understanding of the model's outputs are desirable, such as in the medical and legal fields. Interpretability was not addressed in this research and it's still an under-studied area. Therefore, future research might include developing and applying interpretability techniques to these powerful universal architectures.

VII. Conclusion

In this paper, we have motivated the importance of continuing research in the Artificial Intelligence sub-field of multimodal learning, in particular by presenting some of the current challenges in the field and presenting the literature on modern approaches to address them. We further address these challenges by exploring the Perceiver architecture and attempting to improve its task performance by integrating a deformable attention module into it. After a successful integration experiments showed that it might be possible to increase performance with the proposed method. Nonetheless, there were also results with slightly decreased performance. This means additional experimentation with other datasets and modalities are needed to make a definite statement about the impact of deformable attention in the Perceiver architecture task performance.

References

[1] Khaled Bayoudh, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa. A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets. *The visual computer*, 38(8):2939–2970, 2022.

[2] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[4] Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, Zheng-Ning Liu, Peng-Tao Jiang, Tai-Jiang Mu, Song-Hai Zhang, Ralph R Martin, Ming-Ming Cheng, and Shi-Min Hu. Attention mechanisms in computer vision: A survey. *Computational visual media*, pages 1–38, 2022.

[5] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021.

[6] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021.

[7] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys*, 54(10s):1–41, 2022.

[8] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[10] https://colab.research.google.com/ (accessed on 09.09.2023).

[11] https://github.com/angelonti/computer-vision-master-project-2022-2023 (accessed on 09.09.2023).

[12] https://huggingface.co/datasets/zh-plus/tiny-imagenet (accessed on 09.09.2023).

[13] https://huggingface.co/deepmind/vision-perceiver-learned (accessed on 09.09.2023).

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[15] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021.

[16] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4794–4803, 2022.

[17] Peng Xu, Xiatian Zhu, and David A Clifton. Multimodal learning with transformers: a survey. *arXiv preprint arXiv:2206.06488*, 2022.

[18] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021.

[19] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

VIII. List of Acronyms

CNN Convolutional Neural Network
RNN Recurrent Neural Network
SOTA state-of-the-art
MLP Multilayer Perceptron
LN Layer Normalization
FC Fully Connected
DAT Deformable Attenion Transformer